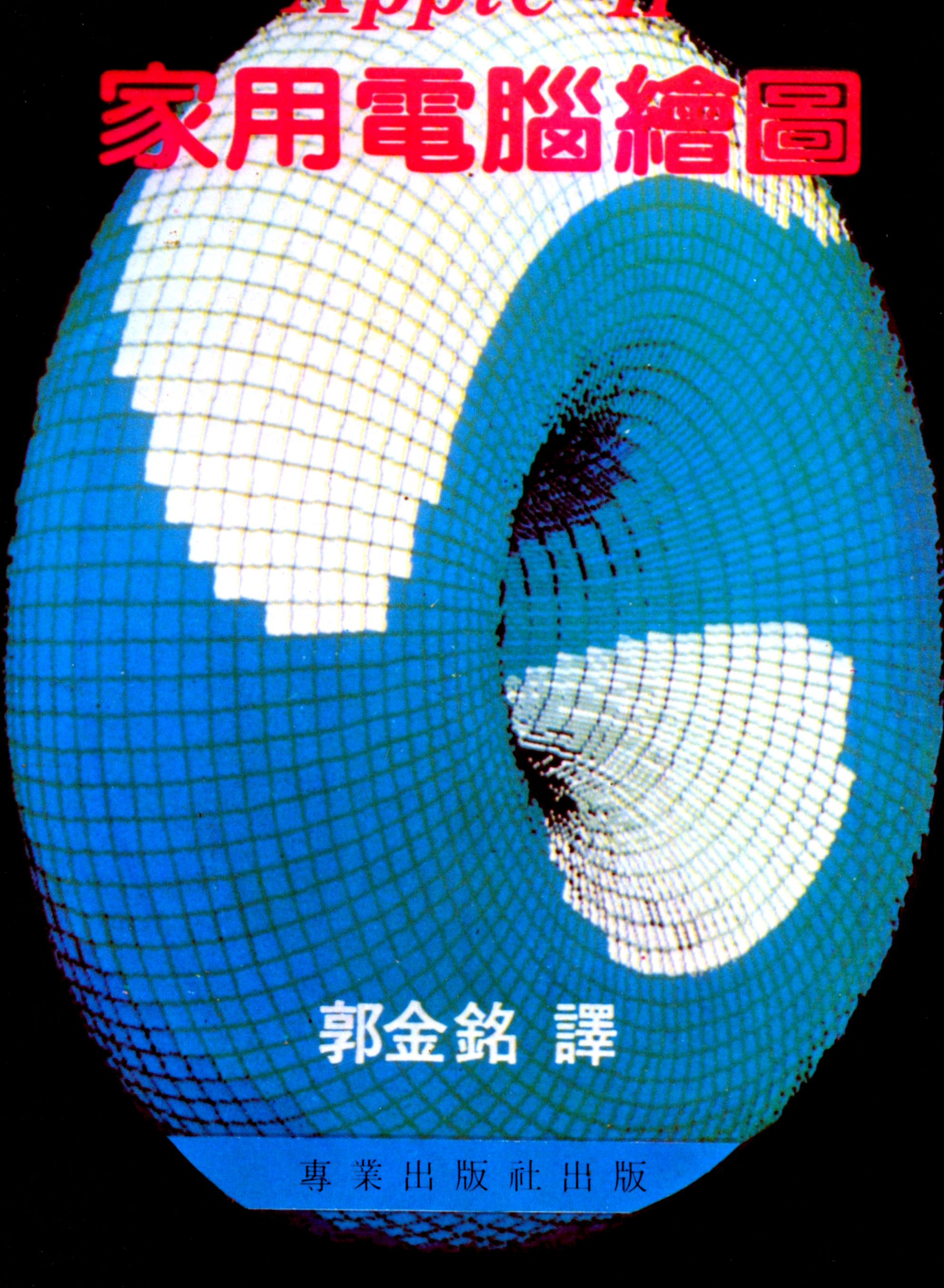
Apple II

家用電腦繪圖



郭金銘 譯

專 業 出 版 社 出 版

Apple II
家用電腦繪圖

郭金銘 譯

專 業 出 版 社 出 版

AppleII
家用電腦繪圖

編譯者：郭金銘

出版兼
發行者：專業出版社

九龍大角咀橡樹街83６號

承印者：聯基印刷廠

香港仔偉倫工業大廈九樓

定價：H.K.$25.00

前　言

雖然現在美國Apple II軟體中有關繪圖的程式組或專用語言，如LOGO及VISIPLOT等逐漸佔有市場，但Graham 先生和我本人基於對Apple II的興趣和見解，寫出了這本獨立而不與任何專用繪圖系統有關的〝電腦繪圖〞一書。

此書利用一種新的Videocomp electronic typesetting 機器來顯示全部的圖形範例，使你覺得耳目一新。本書的另一特性是易讀易懂，步步分析的方法使讀者能在自己的電腦上達成簡單的圖形效果，同時可視讀者的興趣與需要而以最少的精神使其發展出更多複雜的圖形效果，希望讀者讀此書時，能分享作者群在發展這些繪圖技巧和程式時的成就與長時的快樂，使你能在自己的Apple II 上畫出任何想像的圖形。

此書中各程式是在48K的Apple II Plus發展的，程式集磁碟片為TYCON 公司所發行，地址是TYCON Corporation，CN 5258，Princeton，N.J.，08540。

（譯者按：本書各節後的建議練習是對讀者的挑戰，如果你能依次將各節的理論應用到各練習題中，則必將受益良多）。

前　言

第一章　本書的內容 1

第二章　畫些簡易圖形 5

2-1　用低解析度畫一張臉 5
2-2　基本動態處理 10
2-3　頑皮狗 13
2-4　高解析度繪圖 18
2-5　蜈蚣圖 23
2-6　將圖按步畫出 27

第三章　建立圖形塊和圖形表 33

3-1　南太平洋島嶼圖 33
3-2　能保存的圖景 39
3-3　建立圖形塊 47

3-4 DESIGN 2程式 …… 51
3-5 使用Apple 圖形表 …… 60
3-6 STATIC 2 程式 …… 67
3-7 奇怪的方形 …… 70
3-8 鐵 鎚 …… 73

第四章 畫 圖 …… 77

4-1 如何畫圖 …… 77
4-2 小精靈 …… 82
4-3 SON OF MR. MOUTH程式 …… 87
4-4 方形和其他幾何圖形 …… 90
4-5 心臟、腎臟和其他圖形 …… 92

第五章 統計曲線 …… 103

5-1 商用圖表 …… 103
5-2 建立畫圖用的文字檔 …… 104
5-3 簡易曲線程式 …… 116
5-4 長條圖 …… 130
5-5 比率圖 …… 142

第六章 曲線圖形 …… 153

6-1 抛物體軌跡 …… 153
6-2 上下移動的曲線 …… 159
6-3 用點來決定多項式曲線 …… 180
6-4 使角平滑的曲線 …… 189

6-5 BEZEIT程式 …… 195
6-6 橢圓軌道 …… 202
6-7 其他圖形 …… 206
6-8 如何畫波形 …… 211

第七章 更深入的應用 …… 227

7-1 物理投射圖 …… 227
7-2 TRYTOR程式 …… 233
7-3 萬花筒 …… 237
7-4 一些簡易幾何學 …… 239

第八章 Applesoft 語言簡述 …… 241

8-1 APPLE裡的語言 …… 241
8-2 APPLESOFT程式 …… 242
8-3 初設定及指定 …… 243
8-4 數值及文字陳式 …… 244
8-5 流程控制 …… 246
8-6 I/O指令 …… 248
8-7 繪圖指令 …… 251
8-8 使用磁碟機 …… 255

第一章　本書的內容

無論我們走到那裡，都可以看到Apple II ，辦公市、家庭及學校裡似乎是Apple II 聚集的地方。你可以在它們的螢幕上任意的顯現出圖形，圖片及眼睛所能見到的模型，此功能大大的增加了幫助和娛樂人類的能力。Apple II 以極低廉的價格給予老少學習用電腦來畫圖的途徑。

許多人都想以數據，問題等以圖形的方式，顯示在電腦螢幕來吸引顧客，廠商或學生的注意力。此書將會教你如何來做這些事情，同時書內許多 BASIC 程式對各種應用而言都極爲有用。

- 在Apple II 的低及高解析度上畫圖
- 在螢幕上畫直綫圖形
- 自己做動態畫面處理
- 畫出有趣圖形和長條統計表
- 使用 PEEKs ，POKEs ，文字檔，圖形表及其他Apple II 在繪圖功能上的細步軟體
- 畫出有趣的模型及創造視覺效果
- 用複雜的電腦輔助設計（ CAD ）方法畫出圓滑曲綫
- 在圖形螢幕上建出不同大小，形狀的文字
- 畫出三維分視圖
- 並對有興趣購買繪圖程式組的人提供權衡的評價功能

首先要聲明的是，你並不需要數學基礎，程式自己會負責並能處理各種不同的情況。當我們介紹使用任何一種公式時，你不一定要去了解它們，不過此書仍將做些解釋。每章都提供多種你修改程式的方法。

如此對下列各種應用都能有所幫助。

- 商業及其他資料的分析及顯示
- 自己設計遊戲
- 計畫商業及其他制度的行爲
- 對各種不同年齡的學童提供教育數學、科學及其他課題的幫助
- 設計工業產品
- 對喜歡解謎語的人提供一種挑戰

無論你喜歡什麼，你都能從在Apple II 上設計程式而得到些技巧和小主意。同時無論你的興趣在那裡，也不論年齡，讀這本書同時執行這些程式都能使你從中得到快樂和助益。就像蘋果落地使牛頓引發重力定律一樣，即使不需爬上樹去，你也可從自己的 Apple II 得到些實際的東西。

首先圖 1-1 到 1-10爲 Apple II 附隨各種不同組件的簡圖。圖1-10 的設備是用來畫出如圖1-11 的有趣圖形。同時你也可將自己的 Apple II 連接在高品質畫圖機，Vediodisk 元件，語言合成器，實驗用和工業用儀器來監督、控制及達成其他各種不同的應用。

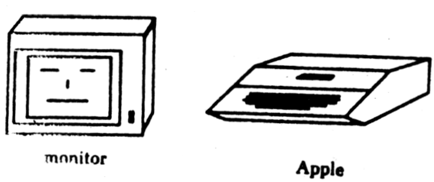


圖 1.1

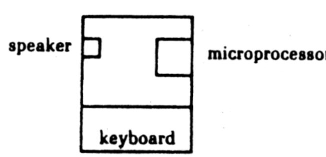


圖 1.2

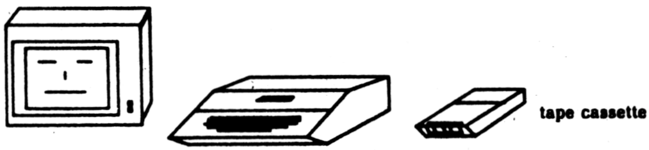


圖 1.3

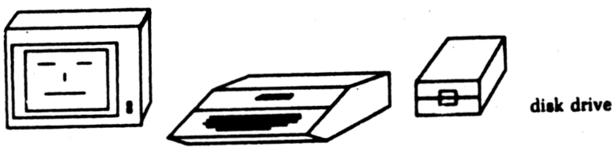


圖 1.4

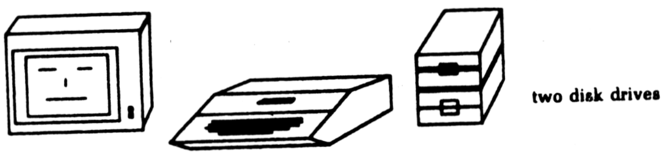


圖 1.5

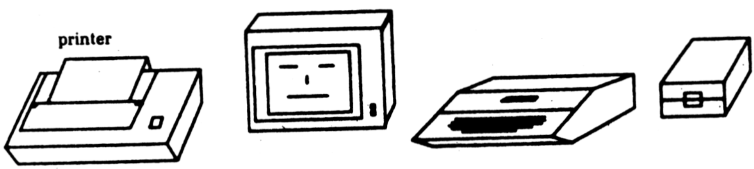


圖 1.6

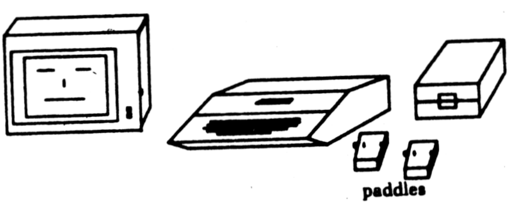


圖 1.7

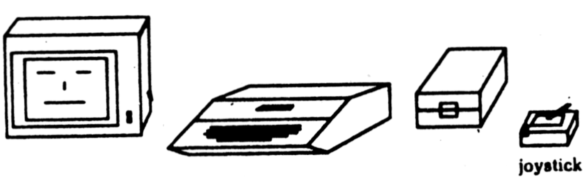


圖 1.8

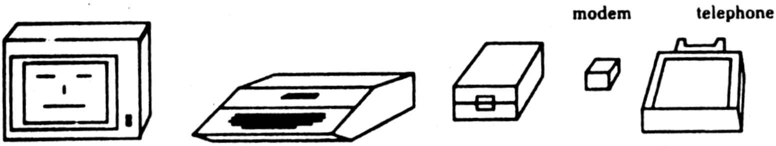


圖 1.9

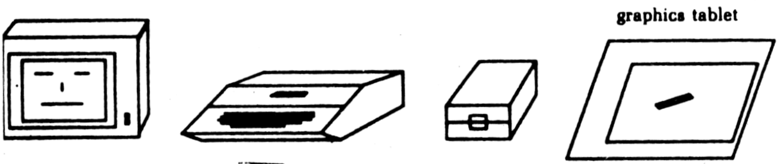


圖 1.10

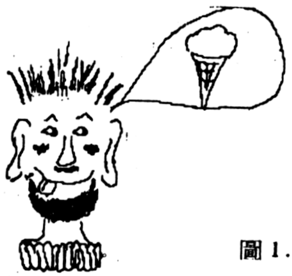


圖 1.11

一個電腦或終端機可藉用調變器連接在電話綫上。調變器是一種將電腦資料轉換成能被電話綫傳送的訊號裝置。藉著調變器和電話綫，一個Apple II可與另一個Apple II〞交談〞——相互傳送資料，圖形及文字等。同時也可使Apple II與其他廠牌的大小電腦〞交談〞。如此Apple II便可用來設計圖形，而這些圖形可存在其他高速的多使用者輸出裝置裡。此書的出版便是將作者在Apple II上畫出的圖形及所寫的程式連接到電子排版打字機上而將其印出。所以無論在商業上，教育上及娛樂上，這種技術的用處極大。

本書裡所有的程式都是用Apple II的Applesoft 寫成的（ BASIC 語言之一種）。對於此語言的詳細書籍各大書店將有出售，如果有需要可以參考一些。第八章則爲對其他種語言有經驗的人做一次簡介以認識Applesoft BASIC 。

第二章　畫些簡易圖形

爲使你能開始用Apple 畫圖，此章以 Applesoft 指令來畫一些簡單的圖形。同時介紹一些高解析度圖形及簡易聲音，並有產生動態效果的方法介紹，最後爲一將各點依其座標連成綫的簡單程式。

2-1　用低解析度畫一張臉

Apple II 螢幕上的圖形可藉由鍵盤打入簡單的指令來產生。圖 2-1 的〞臉〞便是如此。首先將此〞臉〞畫在繪圖方格紙上（如圖 2-2 ）以助解釋指令。

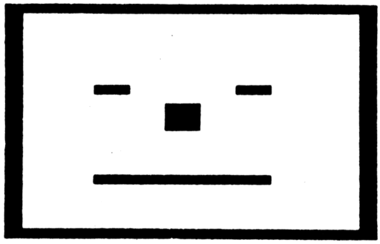


圖 2-1

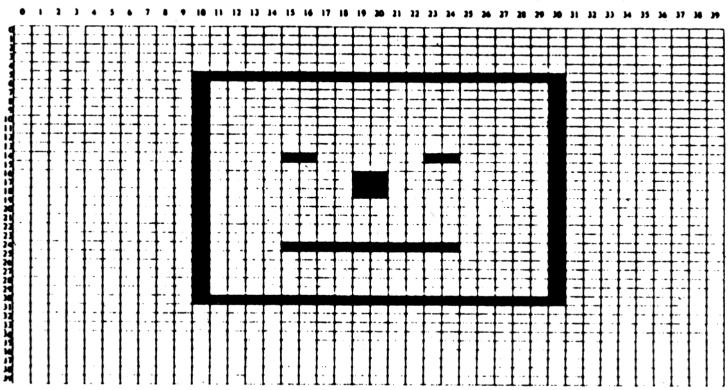


圖 2-2

現在我們依順序介紹畫臉的各指令，第一個指令，

```
GR
```

將電腦設定爲〞低解析度繪圖〞模式，此指令好像將螢幕看做一張每邊有 40 格的繪圖紙，分別以 0 到39編號，如圖 2-2 所示。其中 y 座標由螢幕上方量到下方。第二個指令是，

```
COLOR=15
```

使圖形以白色顯示出來。此對使用螢幕或其他彩色顯示器的效果都一樣。第三個指令是，

```
HLIN 10,30 AT 5
```

在〞臉〞部上端畫一橫綫。HLIN代表〞水平綫〞的英文縮寫。後接的分別是此行的起始點和結束點的 *x* 座標，在 AT 之後的數值代表 *y* 座標。下兩個指令是，

```
HLIN 15,16 AT 14
HLIN 23,24 AT 14
```

將左右兩眼畫出。鼻子則由下列指令畫出。

```
VLIN 16,18 AT 19
VLIN 16,18 AT 20
```

字母VLIN 代表〞垂直綫〞的英文縮寫，其後隨的兩數爲此綫的起始及結束點的 *y* 座標，在 AT 之後的則爲 *x* 座標。嘴部及面部的兩邊及底部則由下列指令畫出。

```
HLIN 15,24 AT 24
VLIN 5,30 AT 10
VLIN 5,30 AT 30
HLIN 10,30 AT 30
```

整組完整的程式爲：

```
GR
COLOR=15
HLIN 10,30 AT 5
HLIN 15,16 AT 14
HLIN 23,24 AT 14
VLIN 16,18 AT 19
VLIN 16,18 AT 20
HLIN 15,24 AT 24
VLIN 5,30 AT 10
VLIN 5,30 AT 30
HLIN 10,30 AT 30
```

如用下列指令則可在嘴邊加上兩個方塊使〝臉〞看起來在笑，即圖 2-3 及 2-4 。

```
PLOT 15,23
PLOT 24,23
```

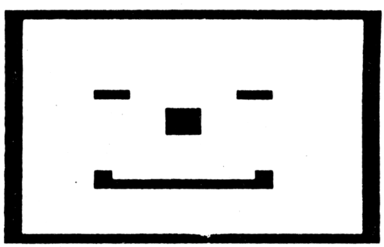


圖 2-3

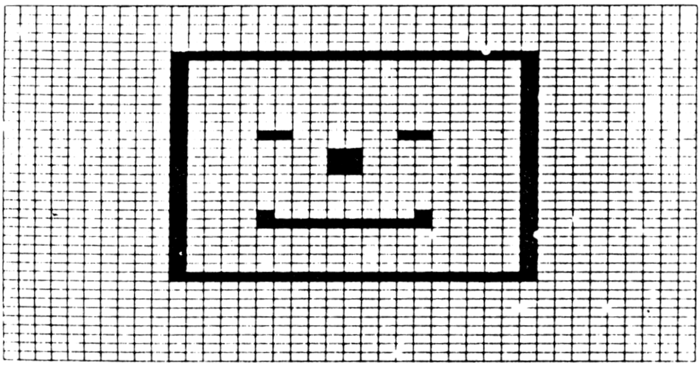


圖 2-4

建議練習

1. 畫一列垂直綫而在頂端對齊。
2. 畫一列垂直綫而在底端對齊。
3. 畫些水平綫段而在(A)左邊(B)右邊對齊。
4. 利用垂直及水平綫畫一些方塊、三角形、菱形及其他幾何圖形。
5. 先在繪圖紙上畫一些其他的圖形，再用指令畫在 Apple 螢幕上。
6. 取一些已在紙上的方形圖，例如磚塊，房子結構圖等，再用指令將其顯示在螢幕上。
7. 如果你在 COLOR 指令裡用其他不同的數目，看看結果如何（即使你用的是黑白顯示器）。

2—2 基本動態處理 (ELEMENTARY ANIMATION)

現在我們再做些更深入的。圖 2-5及2-6 的〞笑臉〞裡的眼睛變大了，每一個眼睛佔據低解析螢幕上的大小為 40 × 40 格的方塊，這些又稱為低解析度光點（pixel）或（brick）。此節尾部的 BASIC 程式WINKFACE能使螢幕上的笑臉左眼一開一閉。

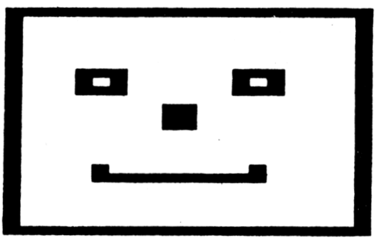


圖 2-5

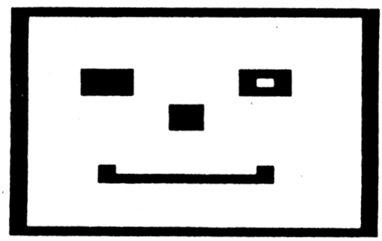


圖 2-6

第 140 到 200 行定義張開眼睛的笑臉，其使用的低解析度繪圖指令，我們都已討論過。第 220 行，

```
220 COLOR=15: PLOT 15,13
```

將左眼閉上一段由第 240 行所決定的時間：

```
240 FOR DE=1 TO 40: NEXT DE
```

眨眼時間的長度可依 FOR 敍述裡的上限值（40）不同而不同。

第260 行，

```
260 COLOR=0: PLOT 15,13
```

將眼球顏色設定爲螢幕的背景顏色（即 COLOR = 0）來反應眼睛的睜閉動作，其時間由第 280 行的 FOR 迴路決定，

```
280 FOR DE=1 TO 40: NEXT DE
```

第 300 行，

```
300 GO TO 220
```

將程式送回第 220 行重新閉上眼睛。

打入 RUN 便可執行此程式，直到按下 CONTROL - C 鍵才停止（亦即，直到同時按下 CONTROL 及 C 鍵才停止）。

前面一節的〞臉〞圖形可分別打入各指令而立即執行之，此因圖形一旦畫出便不再改變。此書裡的大部份圖形都是在執行整個程式後才被畫出。不過，讀者應記住，〞立即執行〞模式在畫草圖時，或對程式結束後的原有圖形添加一些東西是特別有用。

建議練習

1. 使眨動的眼睛不是左眼而是右眼。
2. 使兩眼交互睜閉。
3. 使兩眼同時睜閉。
4. 使嘴部張開並閉上。
5. 使雙眼眨動 100 次。提示：使用一個 FOR 迴路，其起始部份爲，

```
205 F    I=1 TO 100
```

並將第 300 行改爲

```
300 NEXT I
```

6. 以前面各節所提的建議練習中發展一些動態效果。

WINKFACE程式

```
100  REM *** WINKFACE ***
110  REM *** USES LOW-RES GRAPHICS ***
120  GR : COLOR=15
130  REM *** DRAW FACE WITH OPEN EYES ***
140  HLIN 10,30 AT 5: HLIN 14,16 AT 14: HLIN 23,25 AT 14
150  VLIN 16,18 AT 19: VLIN 16,18 AT 20: HLIN 15,24 AT 24
160  VLIN 5,30 AT 10: VLIN 5,30 AT 30: HLIN 10,30 AT 30
170  PLOT 15,23: PLOT 24,23
180  HLIN 14,16 AT 12: HLIN 23,25 AT 12
190  PLOT 14,13: PLOT 16,13
200  PLOT 23,13: PLOT 25,13
210  REM *** CLOSE LEFT EYE ***
220  COLOR=15: PLOT 15,13
230  REM *** TIME DELAY WHILE IT IS CLOSED ***
240  FOR DE=1 TO 40: NEXT DE
250  REM *** OPEN LEFT EYE ***
260  COLOR=0: PLOT 15,13
270  REM *** TIME DELAY WHILE IT IS OPEN ***
280  FOR DE=1 TO 40: NEXT DE
290  REM *** REPEAT THE CYCLE ***
300  GOTO 220
```

2-3 頑皮狗

這裡的 DOGADD 程式就要用到些數學了。螢幕上的光點會伴隨一訊號以問使用者是否要加，減或乘兩數，在使用者回答後，程式再問要做運算的兩數爲何。當輸入這些數之後，狗藉著跨幾次前脚來回答正確的結果。如果結果值超過 100，它便以搖動尾巴來代替超過一百的部份而剩下的仍以踏前脚表示。圖 2-7 A 及 2-7 B 分別顯示此狗在放下前脚及舉起前脚的樣子。

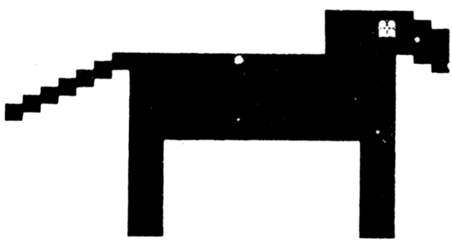


圖 2-7A



圖 2-7B

此程式使用的主要變數為：

X，Y	用來畫點和綫的座標
PL	在DATA敍述裡列出座標的索引
IN$	使用者要求的運算動作字串
W$	其第一字元
A1，A2	做運算的兩數
AN	答案
H	答案中一百的次數（例如，如果答果是245，則H為2）
AM	答案中一百的部份（即H×100）
R	剩下的答案
DE	用來控制一延遲週期的索引
LI	控制踏脚動作的索引
Z	控制錯誤訊息的索引

此程式

1. 在其靜止位置上畫出隻狗。
2. 顯出要求運算的符號，並將此字存放在變數IN$中。
3. 使用字串函數

```
LEFT$
```

來檢驗是否第一字元為A（加法），B（減法）或M（乘法）。然後依結果跳到要求輸入兩數並適當的計算此結果的程式段。
4. 決定答案是否超過100；如是則叫″搖尾″副程式H次。此副程式將用另5個副程式分5次上下動作來搖動尾巴。

5. 決定答案是否恰爲 100 的倍數；如是則狗踏適當次數的前脚以代表答案數。
6. 再重頭開始。

指令

```
COLOR=12      及      COLOR=15
```

需用在畫出身體及眼睛部份之前，以設定不同的顏色，或對黑白螢幕而言設定不同的深淺度。對於顏色的範圍 0-15，你可用 VLIN 指令畫出一列垂直綫段來看出各顏色的代號。

此程式同時也顯示了 HOME，VTAB，HTAB，READ 及 DATA 敍述的使用法。

建議練習

1. 使狗以踏後脚來代替答案的十位數部份，而前脚則代表個位數部份。
2. 用不同的圖形來動態化此結果。例如眨眼的袋鼠，閃爍的狒狒等。
3. 找出一種表現負號的方法，並實行之。
4. 容許使用者打入下列的算術式，而其運算元（即＋，－或×號）在兩

   1+2　　3−2　　2×3

   數之間。〔提示：(1)將整個算術式存到字串中。(2)用個 FOR 迴路和 MID$ 函數來檢查字串中的每個字元。(3)用 LEFT$ 及 RIGHT$ 函數將字串中的兩數分開。(4)用 VAL 函數來求此二數字的數值。〕
5. 讓使用者能以羅馬計數方式打入此二數。
6. 使此狗能對各羅馬數字做些反應。（例如，分別對 I，V 及 X 數字以

閉左眼，閉右眼及閉嘴等動作反應）

7. 使其能以二進位制輸入數目。
8. 使其以二進位的方法表示答案。（例如可用狗的兩個肉峯代表二進位制）
9. 使其能接受十六進位制數目。
10. 擴充動物的範圍以處理如時間，距離，面積，金額及其他不同單位的數字。
11. 此對具數學頭腦的人是項挑戰——使動物能處理分數。〔提示：利用最大公約數法〕

DOGADD程式

```
100  REM *** DOGADD ***
110  REM *** USES LOW-RES GRAPHICS ***
120  HOME : GR : COLOR=12
130  REM *** DRAW THE DOG ***
140  FOR X=10 TO 11: VLIN 35,25 AT X: VLIN 35,25 AT X+13: NEXT X
150  FOR Y=24 TO 15 STEP -1: HLIN 10,24 AT Y: NEXT Y
160  FOR PL=1 TO 14: READ X,Y: PLOT X,Y: NEXT PL
170  DATA 9,15,9,16,8,16,8,17,7,17,7,18,6,18
180  DATA 6,19,5,19,5,20,4,20,4,21,3,21,3,22
190  FOR Y=10 TO 14: HLIN 21,25 AT Y: NEXT Y
200  VLIN 11,15 AT 26: VLIN 12,16 AT 27
210  COLOR=15: PLOT 24,12: PLOT 24,11
220  REM *** PROMPT FOR INPUT ***
230  INPUT "ADD, SUBTRACT OR MULTIPLY? ";IN$
240  INPUT "NUMBERS TO BE COMBINED: ";A1,A2
250  REM *** COMPUTE THE RESULT ***
260  W$=LEFT$(IN$,1)
270  IF W$="A" THEN AN=A1+A2: GOTO 310
280  IF W$="S" THEN AN=A1-A2: GOTO 310
290  IF W$="M" THEN AN=A1*A2: GOTO 310
300  GOSUB 800: GOTO 230: REM *** BAD INPUT ***
```

```
310  IF AN>0 THEN 360: REM *** BRANCH IF POSITIVE ***
320  IF AN<0 THEN  GOSUB 800: GOTO 230: REM *** BRANCH IF NEGATIVE *
330  HOME : VTAB 21: HTAB 10: PRINT "0": REM *** RESULT IS 0 ***
340  FOR DE=1 TO 100: NEXT DE: GOTO 670
350  REM *** SPLIT THE RESULT INTO DIGITS ***
360  H=INT(AN/100):R=AN-H*100
370  IF H=0 THEN 580: REM *** NO HUNDREDS ***
380  REM *** WAG THE TAIL FOR HUNDREDS ***
390  FOR LO=1 TO H
400  COLOR=0: GOSUB 690: COLOR=12: GOSUB 710
410  FOR DE=1 TO 5: NEXT DE
420  COLOR=0: GOSUB 710: COLOR=12: GOSUB 730
430  FOR DE=1 TO 100: NEXT DE
440  COLOR=0: GOSUB 730: COLOR=12: GOSUB 750
450  FOR DE=1 TO 10: NEXT DE
460  COLOR=0: GOSUB 750: COLOR=12: GOSUB 770
470  FOR DE=1 TO 10: NEXT DE
480  COLOR=0: GOSUB 770: COLOR=12: GOSUB 750
490  FOR DE=1 TO 10: NEXT DE
500  COLOR=0: GOSUB 750: COLOR=12: GOSUB 730
510  FOR DE=1 TO 10: NEXT DE
520  COLOR=0: GOSUB 730: COLOR=12: GOSUB 710
530  FOR DE=1 TO 10: NEXT DE
540  COLOR=0: GOSUB 710: COLOR=12: GOSUB 690
550  HOME : VTAB 22: HTAB 10: PRINT LO*100
560  PRINT CHR$(7);: REM *** BELL CODE FOR BEEP ***
570  NEXT LO:AM=100*H
580  IF R=0 THEN 670: REM *** NO UNITS ***
590  REM *** FOOT STAMPS FOR UNITS ***
600  FOR LO=1 TO R: COLOR=0: FOR LI=35 TO 30 STEP -1
610  HLIN 23,24 AT LI: FOR DE=1 TO 5: NEXT DE,LI
620  PRINT : COLOR=12: FOR LI=30 TO 35: HLIN 23,24 AT LI
630  FOR DE=1 TO 5: NEXT DE,LI: HOME : VTAB 22
640  PRINT CHR$(7);: REM *** BELL CODE FOR BEEP ***
650  HTAB 10: PRINT LO+AM: NEXT LO
660  REM *** REPEAT CYCLE ***
670  GOTO 230
680  REM *** DRAW TAIL IN DIFFERENT POSITIONS ***
```

```
690  RESTORE : FOR PL=1 TO 14: READ X,Y: PLOT X,Y: NEXT PL
700  RETURN
710  HLIN 9,7 AT 15: HLIN 9,7 AT 16: HLIN 6,4 AT 17: HLIN 6,4 AT 18
720  RETURN
730  HLIN 9,4 AT 15: HLIN 9,4 AT 16
740  RETURN
750  HLIN 9,7 AT 16: HLIN 9,7 AT 15: HLIN 6,4 AT 14: HLIN 6,4 AT 13
760  RETURN
770  PLOT 9,15: PLOT 9,16: PLOT 8,15: PLOT 8,16: PLOT 7,15: PLOT 6,15
780  PLOT 7,14: PLOT 6,14: PLOT 5,14: PLOT 5,13: PLOT 4,13: PLOT 4,14
790  RETURN
800  REM *** USER GAVE INVALID INPUT ***
810  FOR Z=1 TO 3: FOR X=6 TO 0 STEP -1
820  COLOR=3: VLIN 39-X*3,40-X*3 AT 3+X
830  COLOR=0: VLIN 39-3*X,40-3*X AT 3+X
840  FOR DE=1 TO 4: NEXT DE
850  NEXT X,Z
860  RETURN
```

2—4 高解析度繪圖

ROBOT 程式顯示出一個小 Robot 位在箭頭的行徑綫上（參見圖 2-8）。當一箭擊中它的脚時，Apple 的喇叭發出一短聲而脚的部份消失。當箭擊中頭部，則 Robot 被殺。然後出現另一隻 Robot 繼續此程序，直到你按下 CONTROL - C 爲止。各 Robot 和箭的垂直位置爲隨機產生的。

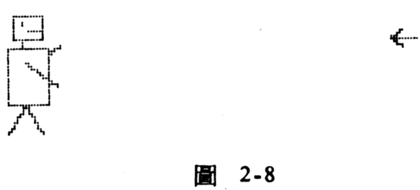


圖 2-8

此程式使用 Apple 的高解析繪圖模式。此模式的螢幕上半部爲一組方

形點列陣，而光點的面積較低解析度的要小的多。在高解析度模式裡，橫的有 280 個點（編號 0 到 279 ），縱的有 160 個點（編號 0 到 159 ），最後底端留 4 條文字行。同時上面的繪圖區也可擴充到整個螢幕，而使其共有 280 × 192 個點。

此程式裡使用的主要變數是：

XR，YR	Robot 頭上左上點的座標
X，Y	箭尾的座標
DE	延遲連續箭位置的 FOR 迴路索引
H	地址 − 16336 ，用在 PEEK 指令裡使喇叭發出聲響

此程式

1. 利用隨機數產生函數 RND 計算一在 0 與 1 之間的數，再用 INT 函數將此數乘以60後的整數部份取出，做爲 Robot 的座標，即 XR，XY。此陳式爲

```
INT(RND(1)*60)+20
```

所以能取出在20到80之間的隨機數。RND 函數的參數必須是個正整數以產生不同的隨機數，但其本身值却不重要。

2. 用指令

```
HGR: HCOLOR=3
```

轉到高解析繪圖模式，並在顯示器上以白色顯示圖片。

3. 跳到一個畫出 Robot 頭及身體的副程式，並跳到另一畫出其四肢的

副程式。

4. 再次使用 RND 函數及 INT 函數在 0 到80之間取出一隨機整數，此即爲箭的 *y* 座標。注意此範圍較 Robot 的範圍大。此陳式爲：

```
Y=YR+INT(RND(1)*80)-20
```

5. 以螢幕上連續 8 個點顯示表箭的圖形。使用 FOR 迴路，其中每一圈內指令，

```
HCOLOR=3: GOSUB 550
```

將在一新位置顯示箭，而指令，

```
X=X+10: HCOLOR=0: GOSUB 550
```

將其自原位置擦去。

6. 決定是否箭擊中 Robot 的身體，頭部或四肢，並適當的做轉移。

7. 繼續箭的向左飛行運動，直到碰及螢幕左邊而消失，並重覆發箭的過程。

8. 由指令：

```
FOR LO=1 TO 30: X=PEEK(H): NEXT LO
```

產生一聲長鳴的前一部份（如果直接擊中）。此指令用PEEK 功能將喇叭連續響30次，而其函數參數爲－ 16336 。對此功能我們等會兒再討論。同理可將喇叭重覆 3 次，每次20響，使其產生另一種不同的聲調。迴路的最後一個敍述回到此處理程序再重頭開始。

9. 如果擊中四肢便將其擦去，方法是將畫四肢的副程式中顏色部份設定爲背景顏色。然後產生一短鳴，同樣的是用PEEK 功能。接著是

由空PRINT敍述造成一短暫延遲。

建議練習

1. 使Robot 以小塊的方式消失。
2. 列出一列 Robot 而逐次被擊中而消失。
3. 使兩個 Robot 面對面出現，而被擊中後交換位置。
4. 在螢幕一邊顯示出一塔而由一連串投射物擊中而逐漸消失。
5. 在文字區內，顯示出對現存的 Robot 所發出的箭數。
6. 同時也顯示出將 Robot 打死所需的平均箭數。
7. 容許使用者改變箭束的垂直範圍，並看看此改變與除去一隻 Robot 所需的平均箭數有何關係。
8. 同時也容許使用者改變 Robot 的大小，並觀察此與平均數有何影響。

ROBOT程式

```
100  REM *** ROBOT ***
110  HOME : REM *** CLEARS THE SCREEN ***
120  H=-16336: REM *** PEEKVALUE FOR CLICK ***
130  REM *** USES HI-RES GRAPHICS ***
140  HGR : HCOLOR=3
150  REM *** DRAW A ROBOT ***
160  XR=20:YR=INT(RND(1)*60)+20
170  GOSUB 410: GOSUB 510
180  REM *** FIRE AN ARROW ***
190  Y=YR+INT(RND(1)*80)-20
200  FOR X=200 TO 40 STEP -20
210  HCOLOR=3: GOSUB 550
220  HCOLOR=0:X=X+10: GOSUB 550
230  FOR DE=1 TO 7: NEXT DE
```

```
240  NEXT X
250  IF Y>YR AND Y<YR+30 THEN 340: REM DIRECT HIT
260  IF Y>YR AND Y<YR+40 THEN 370: REM HIT ON LEGS
270  FOR X=30 TO 10 STEP -20: REM *** MISSED ***
280  HCOLOR=3: GOSUB 550
290  X=X+10: HCOLOR=0: GOSUB 550
300  FOR DE=1 TO 7: NEXT DE
310  NEXT X
320  X=10: GOSUB 550: GOTO 190
330  REM *** GROAN ***
340  FOR LO=1 TO 30:X=PEEK(H): NEXT LO
350  FOR LO=1 TO 20:X=PEEK(H)+PEEK(H)+PEEK(H): NEXT
360  GOTO 140
370  HCOLOR=0: GOSUB 510: FOR LO=1 TO 10
380  X=PEEK(H)+PEEK(H): PRINT :X=PEEK(H): NEXT LO
390  GOTO 270
400  END
410  REM *** DRAW HEAD AND BODY ***
420  HPLOT XR,YR TO XR+10,YR TO XR+10,YR+8 TO XR,YR+8 TO XR,YR
430  HPLOT XR+3,YR+2 TO XR+3,YR+3
440  HPLOT XR+5,YR+5 TO XR+10,YR+5
450  HPLOT XR+3,YR+8 TO XR+3,YR+11
460  HPLOT XR-2,YR+11 TO XR+12,YR+11 TO XR+12,YR+30
470  HPLOT XR+12,YR+30 TO XR-2,YR+30 TO XR-2,YR+11
480  HPLOT XR+4,YR+16 TO XR+15,YR+24
490  HPLOT XR+12,YR+13 TO XR+16,YR+10
500  RETURN
510  REM *** DRAW LEGS ***
520  HPLOT XR+4,YR+30 TO XR-2,YR+40
530  HPLOT XR+4,YR+30 TO XR+10,YR+40
540  RETURN
550  REM *** DRAW ARROW ***
560  HPLOT X,Y TO X-10,Y: HPLOT X-6,Y-3 TO X-10,Y TO X-6,Y+3
570  RETURN
```

2—5 蜈蚣圖

CENTIPEDE 程式如圖 2-9 畫出許多小蜈蚣，並使其跑過螢幕。

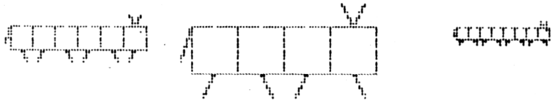


圖 2-9

其中第一隻蜈蚣有 6 隻脚，尾部有一尾巴，前端頭點則有眼睛和觸角（參見圖 2-10）。第二隻和第三隻蜈蚣則分別有 4 隻及 8 隻脚。



圖 2-10

此程式

1. 指引使用者引導蜈蚣的細步圖和其路徑。
2. 在螢幕左邊畫出隻蜈蚣。
3. 重覆的將此蜈蚣的尾段，尾巴，眼睛和觸角擦去，然後在尾段畫出尾巴，並在前面加上一段，再在此段內畫上眼睛和觸角。
4. 如果蜈蚣撞到頭則停止。

指引訊息需爲：

1. 一段大小，此必需爲光點除以 4 之後的每邊長（應爲整數）。
2. 一隻蜈蚣的段數。
3. 走多少步才能橫過螢幕。
4. 蜈蚣在螢幕上的垂直位置。

一個典型的數組（符合上面要求）爲 2 ，5 ，25 ，60 。

此圖所需的幾何知識相當簡單（參見圖 2-11）。程式裡用到的變數爲:

A	各段的大小（即每段的 1 ／ 4 長）
SM	段數
PM	步數（參見程式分析第 3 點）
YO	蜈蚣背部的 *y* 座標（即各段的頂部）
XO	正在處理（畫或擦）段的左上角 *x* 座標
XT	尾段的左上角 *x* 座標
FT	各段左邊與脚的距離（由 A 到 3A 不等）

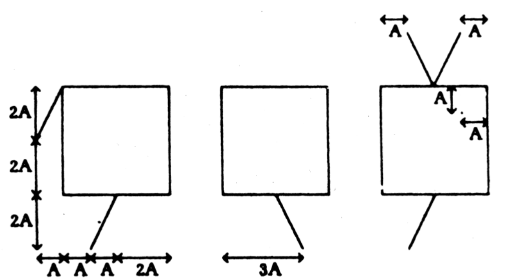


圖 2-11

程式使用一個副程式將一段擦去或畫上，第二個副程式則做畫或擦尾巴的工作，第三個是畫擦眼睛和觸角。

建議練習

1. 蜈蚣起始和結束在螢幕內。
2. 可容許使用者改變前進速度。
3. 速度可改變所需步數。
4. 由右橫渡到螢幕左邊。
5. 在螢幕上上下下移動的蜈蚣。
6. 當一蜈蚣走完後，出現另一隻蜈蚣，其大小速度及方向由隨機函數產生。
7. 數隻蜈蚣同時在螢幕上一起前進或做前進競賽。
8. 記錄蜈蚣移動頭部的次數並相對改變訊息。
9. 在使用者打入所需資料後，程式要求(A)蜈蚣長度，以螢幕單位表之。(B)運動距離並告訴使用者反應是否正確。
10. 利用同樣的原理畫一組圖形橫過螢幕。
11. 畫出一個機器人競賽而第一位應手持旗子。
12. 設計一些其他應用上的動態效果。

CENTIPEDE程式

```
100  REM *** CENTIPEDE ***
110  REM *** PROMPT FOR INPUT ***
120  TEXT : HOME : VTAB 21
130  PRINT "SIZE (CAN BE 1-6, NOW IS ";A;"): ";: INPUT "";A
140  PRINT "LENGTH (CAN BE 1-10, NOW IS ";SB;"): ";: INPUT "";SB:SM=SB*2
150  PRINT "RUN (CAN BE 1-30, IS NOW ";PM;"): ";: INPUT "";PM
160  PRINT "POSITION (CAN BE 10-150, IS NOW ";YO;"): ";: INPUT "";YO
```

```
170  A2=A+A:A4=A2+A2:A3=A2+A
180  XT=A4:X0=A4:X4=X0+A4
190  Y1=Y0+A:Y2=Y0+A2:Y3=Y2+A
200  Y4=Y0+A4:Y6=Y4+A2:YA=Y0-A2:FT=A
210  REM *** DRAW THE CENTIPEDE - START WITH TAIL ***
220  HGR : HCOLOR=3: GOSUB 490
230  REM : DRAW BLOCK 2 FOR ALL THE SEGMENTS
240  FOR SN=1 TO SM
250  GOSUB 520
260  REM : MOVE THE CORNERS AND THE FOOT
270  X0=X4:X4=X0+A4:FT=A4-FT
280  NEXT SN
290  REM *** DRAW THE EYE ***
300  GOSUB 560
310  REM *** MAKE IT RUN ***
320  FOR P=1 TO PM
330  REM *** BRANCH IF ERROR WILL OCCUR ***
340  IF X4>279 THEN 1000
350  REM *** ERASE THE TAIL ***
360  HCOLOR=0: GOSUB 490
370  REM : ERASE THE END SEGMENT
380  XP=X0:X0=XT:X4=X0+A4: GOSUB 520:X0=XP:X4=X0+A4
390  REM *** ERASE THE EYE ***
400  GOSUB 560
410  REM *** DRAW THE NEW TAIL ***
420  HCOLOR=3:XT=XT+A4: GOSUB 490
430  REM *** DRAW THE NEW FRONT SEGMENT AND EYE ***
440  GOSUB 520:X0=X4:X4=X0+A4: GOSUB 560
450  REM *** REVERSE THE DIRECTION OF THE LEG ***
460  FT=A4-FT
470  NEXT P
480  GOTO 120
490  REM *** DRAW BLOCK 1 *** (TAIL & SIDE IT IS ON) ***
500  HPLOT XT-A,Y3 TO XT,Y0 TO XT,Y4
510  RETURN
520  REM *** DRAW BLOCK 2 *** (3 SIDES OF SEGMENT AND LEG)
530  HPLOT X0,Y0 TO X4,Y0 TO X4,Y4 TO X0,Y4
540  HPLOT X0+A2,Y4 TO X0+FT,Y6
550  RETURN
560  REM *** DRAW BLOCK 3 *** (THE EYE AND ANTENNAE) ***
```

```
570  HPLOT X0-A,Y1
580  HPLOT X0-A3,YA TO X0-A2,Y0 TO X0-A,YA
590  RETURN
1000 REM *** ERROR ***
1010 TEXT : HOME : FLASH : VTAB 12: HTAB 10
1020 PRINT "DON'T BANG MY HEAD!": NORMAL
1030 FOR DE=1 TO 1000: NEXT DE: GOTO 120
```

2—6　將圖按步畫出

DRAW1程式使你能在螢幕上逐點畫圖。其在繪圖螢幕邊緣畫出兩軸，而每格10點做一深色記號。整個程式的功能以指示表的方式顯示在文字模式中：

```
 1 MOVE ABSOLUTE
 2 MOVE RELATIVE
 3 DRAW ABSOLUTE
 4 DRAW RELATIVE
 5 ERASE ABSOLUTE
 6 ERASE RELATIVE
 7 CLEAR SCREEN
 8 QUIT
WHICH OPTION?
```

程式的功能正如同你自己做的下列動作：

1. 在螢幕上任意移動筆位置，而不留任何記號。
2. 由筆的起始點和終點間用筆畫一直綫。
3. 將筆改成橡皮擦，並如第 2 點一樣自起點到終點將一綫擦去，

同時你可在MOVE，DRAW 或 ERASE 指令的尾端說明座標的量度法是

1. 由螢幕左上角爲準，或
2. 由執行指令時筆所在的位置爲準。

上述兩種量度方法分別稱爲絕對和相對座標系。

此程式先清除螢幕，畫出兩軸和記號，轉換到文字模式並顯示主指示表。你可藉打入其選擇號碼（例如 3 代表 DRAW ABSOLUTE）而選擇一指令。然後程式轉回到繪圖模式並在螢幕底部顯示一光點以要求使用者輸入下一點的絕對或相對座標。在你反應之後，程式便依你要求的將圖畫出。再轉到文字模式並顯示出主指示表。你可持續的依此畫圖直到滿意爲止。下面是兩個我們用來產生圖形的資料。

1, 140, 80;	4, 40, 0;	4, 0, 40;	4, −40, 0;	4, 0, −40;
4, 20, −20;	4, 40, 0;	4, 0, 40;	4, −40, 0;	4, 0, −40;
2, 40, 0;	4, −20, 20;	2, −40, 40;	4, −20, −20;	2, 40, 0;
4, −20, 20.				

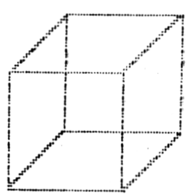


圖 2-12

圖 2-13，一隻火雞，由下列指令順序所產生：

1,　6,　42;	3,　32,　20;	3,　19,　88;	3,　17,　40;	1,　32,　20;
3,　42,　59;	1,　6,　42;	3,　36,　35;	1,　24,　62;	3, 136,　46;
3, 158,　60;	3, 101,　78;	3, 136,　46;	3,　96,　21;	3,　74,　55;
1, 101,　78;	3,　96,　83;	3, 111,　87;	3,　38,　92;	3,　24,　62;
3,　96,　83;	1,　51, 103;	3,　63,　90;	3,　55, 107;	3,　51, 103;
3,　40, 112;	3,　55, 107;	1,　64, 103;	3,　73,　90;	3,　71, 107;
3,　64, 103;	3,　52, 112;	3,　71, 107;	1,　26,　30;	3,　26,　30.

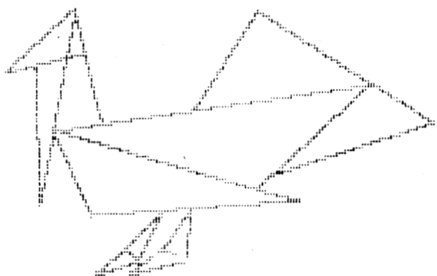


圖 2-13

程式裡用的主要變數為：

XT，YT　在水平及垂直軸記號的 x，y 座標

XO，YO　最近畫的一點 x，y 座標

XN，YN　將要畫的一點 x，y 座標

XC，YC　下一點的相對座標

OP　所說明的選擇號碼

TI$(8)　說明各選擇的字元串

在 Apple 裡各有不同的記憶區來存放你打入文字模式的文字資料及高解析度圖形模式裡的圖形資料。 HGR 指令係先清除此段圖形區並將螢幕與其接上。指令 TEXT 則清除文字資料記憶區並使螢幕接上文字模式記憶區。

此程式將各選擇的說明字元以DATA 敍述內讀到 TI $ 列陣裡，此較使用 8 個指令敍述要短的多。當 Apple 在文字模式時，敍述

```
POKE -16297,0: POKE -16304,0
```

便轉換回高解析度繪圖模式，而不將原在螢幕上的圖形資料清除。

建議練習

1. 容許使用者打入一條綫的長度，並說明是要畫上或是被擦去，以及其與水平的夾角，此爲另一種選擇。
2. 容許使用者擦去兩軸，此爲另一種選擇。
3. 容許使用者將一組定義圖形的數據資料（如圖 2-13上方的數值）以文字檔的方式保存起來，爾後可順序的將資料讀出並畫出各圖形。
4. 在現行筆位置上以一閃爍點或方塊來代表光點（ cursor ）。
5. 使用光點和不同的鍵(A)使其上，下，左，右移動(B)使光點現行位置爲要畫或擦的某條綫的起點(C)使光點現行位置爲此綫之終點（畫綫）(D)使光點現行位置爲此行之終點（擦去），和(E)除去(B)部份的效果，等配合來達到用鍵盤畫圖的目的。
6. 依第 5 項的要求，發展成一完整的鍵盤畫圖程式。
7. 令使用者將資料存到列陣裡，而對各種不同的位置連續取出資料將圖形畫出。
8. 在使用者將資料存到列陣後，使其能對 x ，y 座標分別調整圖形大小

。例如圖2-14裡的兩隻火雞一樣！

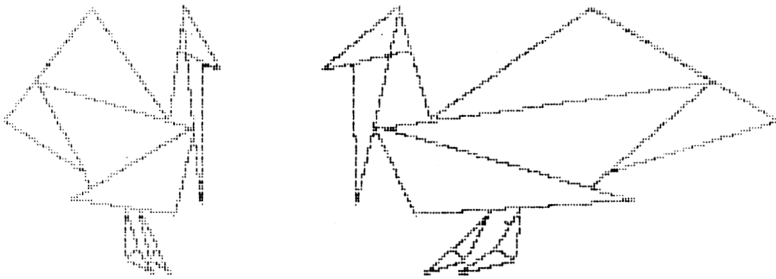


圖 2-14

DRAW 1程式

```
100  REM *** DRAW1-USES ENDPOINT COORDINATES ***
110  LOMEM: 16384: REM *** AVOID MEMORY CONFLICT ***
120  ONERR  GOTO 650: REM *** BRANCH ON ERROR CONDITION ***
130  HOME : HGR : HCOLOR=3: REM *** HI-RES GRAPHICS ***
140  REM *** DRAW AXES AND TICK MARKS ***
150  HPLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0
160  FOR XR=0 TO 279 STEP 10
170  HPLOT XR,0 TO XR,3: HPLOT XR,156 TO XR,159
180  NEXT XR
190  FOR XR=0 TO 279 STEP 50
200  HPLOT XR,0 TO XR,5: HPLOT XR,154 TO XR,159
210  NEXT XR
220  FOR YR=0 TO 159 STEP 10
230  HPLOT 0,YR TO 3,YR: HPLOT 276,YR TO 279,YR
240  NEXT YR
250  FOR YR=0 TO 159 STEP 50
260  HPLOT 0,YR TO 5,YR: HPLOT 274,YR TO 279,YR
270  NEXT YR
280  REM *** INITIALIZE ALL VARIABLES ***
```

```
290  X0=0:Y0=0
300  FOR L=1 TO 8: READ TI$(L): NEXT L
310  DATA MOVE ABSOLUTE,MOVE RELATIVE,DRAW ABSOLUTE,DRAW RELATIVE
320  DATA ERASE ABSOLUTE,ERASE RELATIVE,CLEAR SCREEN,QUIT
330  REM *** DISPLAY MAIN MENU ***
340  TEXT : HOME : HTAB 16: PRINT "OPTIONS"
350  VTAB 4: FOR L=1 TO 8: HTAB 6: PRINT L;".   ";TI$(L): NEXT L
360  PRINT : HTAB 11: INPUT "WHICH OPTION? ";OP
380  IF OP<1 OR OP>8 THEN 340
390  IF OP>6 THEN 620
400  REM *** SWITCH TO HI-RES GRAPHICS DISPLAY ***
410  POKE -16297,0: POKE -16304,0
420  REM *** PROMPT FOR DATA ***
430  VTAB 21: PRINT TI$(OP)
440  PRINT "OLD COORDINATES ARE ";X0;",";Y0
450  IF OP=2 OR OP=4 OR OP=6 THEN 510
460  REM *** ABSOLUTE OPERATION ***
470  INPUT "CHANGE COORDINATES TO ";XN,YN
480  IF XN>=0 AND XN<=279 AND YN>=0 AND YN<=159 THEN 550
490  PRINT "ILLEGAL VALUE(S)": FOR LO=1 TO 1000: NEXT LO
500  VTAB 23: CALL -958: GOTO 450
510  REM *** RELATIVE OPERATION ***
520  INPUT "CHANGE COORDINATES BY ";XC,YC
530  XN=X0+XC:YN=Y0+YC
540  GOTO 480
550  REM *** DRAW OR ERASE A LINE ***
560  IF OP<3 THEN 590
570  HCOLOR=3*(OP<5)
580  HPLOT X0,Y0 TO XN,YN TO X0,Y0
590  X0=XN:Y0=YN
600  FOR DE=1 TO 1000: NEXT DE
610  GOTO 340
620  REM *** CLEAR SCREEN OR QUIT ***
630  IF OP=7 THEN  CLEAR : RESTORE : GOTO 130
640  IF OP=8 THEN  END : REM *** TERMINATES PROGRAM OPERATION ***
650  RESUME
```

第三章　建立圖形塊和圖形表

此章說明一些簡單的方法來定義，顯示和保存圖形的部份資料，並說明如何用此資料重組成複雜的圖形。前二節的程式設計技巧也可用在其他具有層次性的問題設計中。後面各節則說明並使用 Apple 特有的圖形表（ Shape Table ）。

3－1　南太平洋島嶼圖

ALOHA 1 程式畫出島嶼的圖形，裡面可有棕櫚樹和小島（參見圖 3-1 及圖 3-2 ）。你也可用此程式的技巧來畫出其他圖形。



圖 3-1

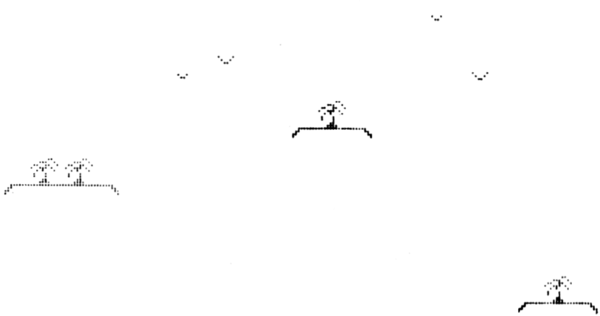


圖 3-2

每一個ALOHA 1裡的畫景包含 5 個你可重覆安排的圖形塊，分別是圖 3-3 裡的小鳥〔圖形塊 1 〕，大鳥〔圖形塊 2 〕，島嶼的左邊〔圖形塊 3 〕和右邊〔圖形塊 4 〕，最後是中間的棕櫚樹〔圖形塊 5 〕。



圖 3-3

此程式將螢幕清除後印出下列訊息以提供使用者選擇：

```
OPTIONS:  1  DISPLAY A BUILDING BLOCK
          2  ERASE SCREEN
          3  QUIT
WHICH OPTION?
```

當時打入 1 時，程式則顯示出，

1. 此數目代表的圖形塊〔亦即，1 代表小鳥，2 代表大鳥等〕。
2. 與最近畫上的圖形塊之間的水平距離〔或剛清除完螢幕時與螢幕左邊的距離〕和此新圖形塊的左緣。
3. 其在螢幕上的垂直距離。

圖 3-1 的產生可先打入 1，後面則隨意取下列三個數目組交換輸入即可。

3, 0, 70;	5, 0, 70;	5, 0, 70;	4, 0, 70;	2, 5, 50;	1, 5, 60;
3, 20, 80;	5, 0, 80;	5, 0, 80;	0, 80;	4, 0, 80;	3, 15, 60;
4, 0, 60;	1, 10, 40;	1, 12, 20;	2, 10, 15.		

前四組（分別含三個數目）顯示第一個島嶼。次兩組，顯示前兩隻鳥。次五組，顯示第二個島嶼，再過來的兩組則顯示無樹的島嶼，最後三組顯示的是最後三隻鳥。

兩圖形塊的水平間距離可爲負數。所以一個圖形塊可畫在另一塊上，如圖 3-4 所示，此圖同時也有山和谷。



圖 3-4

此程式含不同的副程式來畫出各圖形塊，其主要變數爲

ID	指明某一圖形塊的數目
XR	圖形塊左緣的 *x* 座標
YR	圖形塊上緣的 *y* 座標
HS	二圖形塊之間的水平間隔

當你打入顯示一圖形塊的選擇後，程式便要求輸入 ID，HS 及 YR的值並將其保存起來。然後程式依 ID 值跳到不同的副程式。每一副程式內爲連續的 HPLOT 敍述而適當的將圖形塊在指定的地點畫出。XR 值再加上圖形塊的寬度。

要寫一個這樣的程式，需先在紙上將各圖形塊畫好，然後再依圖設計 HPLOT 敍述（參見圖 3-5A 到 3-5E）。



圖 3-5A

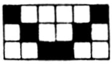


圖 3-5B

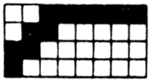


圖 3-5C

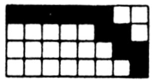


圖 3-5D

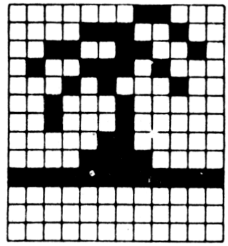


圖 3-5E

此圖形塊的設計工具如果用我們在後面一節的DESIGN 1 程式直接在螢幕上做會快的多。藉著改變副程式，你也可以改變如ALOHA1程式的各圖形塊。第3-4 節裡的DESIGN 2 程式則可使你不需修改程式本身而改變其圖形塊，但此程式要長的多。Apple 裡的另一種特性圖形表，則提供你另一種畫出類似圖形的方法，在某些特定情況下，這種方法的效率很可觀。圖形表將在第 3-5 節到 3-8 節裡介紹，無論如何，多看一些對同一種問題的不同解決方法是很有用的。

建議練習

1. 擴充ALOHA1程式以包括不含樹及島嶼兩斜岸的島嶼，同時也包括船及茅舍等。
2. 設計一些其他的圖景的圖形塊並依此寫出類似ALOHA1 的畫圖程式。
3. 容許使用者能擦去已存在螢幕上的某一圖形塊。
4. 當水平間隔和垂直位置固定時，容許使用者直接輸入欲畫出的圖形塊數目和相對的識別號碼。
5. 給定一個垂直位置和水平距離，試將各連續的圖形由螢幕上方順序顯

示出來。

ALOHA1程式

```
100  REM *** ALOHA1-DRAWS SOUTH SEAS PANORAMA ***
110  REM *** USES HI-RES GRAPHICS ***
200  HGR : HCOLOR=3
210  HOME : VTAB 21
220  PRINT "1 DISPLAY A BUILDING BLOCK": PRINT "2 ERASE SCREEN"
230  PRINT "3 QUIT"
240  HTAB 6: INPUT "WHICH OPTION? ";IZ
250  ON IZ GOTO 300,200,4000
260  GOTO 210
300  REM *** DISPLAY A BUILDING BLOCK"
310  HOME : VTAB 21
320  INPUT "ID#, HORIZONTAL SPACE, AND VERTICAL POSITION: ";ID,HS,YR
330  XR=XR+HS
340  ON ID GOSUB 400,500,600,700,800
350  GOTO 210
400  REM *** DISPLAY A SMALL BIRD ***
410  HPLOT XR,YR-1: HPLOT XR+1,YR TO XR+2,YR: HPLOT XR+3,YR-1
420  XR=XR+4: RETURN
500  REM *** DISPLAY A LARGE BIRD ***
510  HPLOT XR,YR-2: HPLOT XR+1,YR-1: HPLOT XR+5,YR-2
520  HPLOT XR+2,YR TO XR+3,YR: HPLOT XR+4,YR-1
530  XR=XR+6: RETURN
600  REM *** DISPLAY THE LEFT SIDE OF AN ISLAND ***
610  HPLOT XR,YR TO XR+2,YR-3 TO XR+7,YR-3
620  XR=XR+8: RETURN
700  REM *** DISPLAY THE RIGHT SIDE OF AN ISLAND ***
710  HPLOT XR,YR-3 TO XR+5,YR-3 TO XR+7,YR
720  XR=XR+8: RETURN
800  REM *** DISPLAY A PALM TREE ***
805  HPLOT XR,YR-3 TO XR+11,YR-3
810  HPLOT XR+4,YR-4 TO XR+7,YR-4
815  HPLOT XR+5,YR-5 TO XR+6,YR-5
820  HPLOT XR+6,YR-6: HPLOT XR+6,YR-7
```

```
825  HPLOT XR+2,YR-6: HPLOT XR+2,YR-7
830  HPLOT XR+3,YR-8: HPLOT XR+5,YR-8
835  HPLOT XR+9,YR-8: HPLOT XR+1,YR-9
840  HPLOT XR+4,YR-9 TO XR+6,YR-9: HPLOT XR+8,YR-9
845  HPLOT XR+2,YR-10 TO XR+3,YR-10
850  HPLOT XR+6,YR-10 TO XR+7,YR-10
855  HPLOT XR+10,YR-10: HPLOT XR+4,YR-11 TO XR+6,YR-11
860  HPLOT XR+9,YR-11: HPLOT XR+7,YR-12 TO XR+8,YR-12
870  XR=XR+12: RETURN
880  HPLOT XR+4,YR-9 TO XR+6,YR-9: HPLOT XR+8,YR-9
890  HPLOT XR+2,YR-10 TO XR+3,YR-10
4000 END
```

3—2　能保存的圖景

在ALOHA1程式裡，你必須將所畫的各圖形塊以數字的形式輸入電腦而將其畫出。本節的ALOHA2程式則容許你將這些資料保存在文字檔（text file）裡而可以重覆的執行。每當你想要畫圖時便讓電腦將圖形塊資料讀入即可。

保存資料到磁碟片裡的文字檔裡是件極為重要的工作。例如，當你處理

1. 經常需修正，總和，和拿取的商業資料。
2. 在〞交談式〞（ interactive ）設計發展中的資料。
3. 一個程式的輸出，而你需要用來做為另一種程式的輸入資料。
4. 數個程式同時需要的大量常數資料。

ALOHA2程式的最後一部份也可用來做為上述這些應用的範例。

此程式首先顯示出：

```
1. CREATE A PANORAMA
2. SAVE A PANORAMA
3. READ A PANORAMA
4. QUIT
WHICH OPTION?
```

正如同使用ALOHA1程式一樣，要畫一圖景，打入1，接著便得到第二個指令表：

```
1. DISPLAY BUILDING BLOCK
2. ERASE SCREEN
3. RETURN TO MAIN MENU
```

正如前面所用的程序一樣，每次你打入1，便得到：

```
ID #, HORIZONTAL SPACE, AND VERTICAL POSITION:
```

在你輸入這些資料後，螢幕上便顯示一圖形塊。再打入2則可開始畫圖景。當你想存此圖景時，打3以回到主指示表。然後打2，則螢幕上顯出：

```
NAME OF FILE TO BE SAVED:
```

你能

1. 輸入任何一個現存於磁片上而現在未被用到的檔名。
2. 輸入一個已存在的檔名（如此自然將原檔蓋掉）。
3. 將資料附加在一已存在的檔案後面。

打入了你要的檔名後，程式便再問：

```
TYPE B TO BEGIN OR A TO APPEND:
```

依你自己的意願打入A或B便可將現行的圖景存起來。程式繼續回到主指示表，而你可依意願而繼續做選擇。

當一圖景被存到磁片上後，如果在主指示表裡打入3，則程式自動將此圖景顯示出來，然後顯示：

```
NAME OF FILE TO BE READ:
```

你打入適當的檔名後，程式每顯示出一圖景後暫停一下，將其擦去後再繼續畫出下一圖。

ALOHA2 程式裡用到的變數為：

OP	主指示表選擇數
IZ	次指示表選擇數
ID	圖形塊識別號碼
HS	水平距離
XR	圖形塊 x 值
XY	圖形塊 y 值
ID(100)	在圖景中連續圖形塊的儲存列陣
HS(100)	在圖景中連續圖形的 HS 列陣
YR(100)	在圖景中連續圖形的 YR 列陣
IB	上述各列陣的索引變數。在完成整個圖景後 IB 內含值即總圖形塊數。
LO	將圖形塊資料寫入一文字檔時對上述列陣的索引變數。
RE	將圖形塊資料讀出一文字檔時上述列陣的索引變數。

TI$	讀或寫的文字檔名
DE	讀入一圖景時，FOR延遲迴路內的迴路變數
D$	CONTROL-D碼—此為ASCII值4

變數IZ，ID，HS，XR及YR和在ALOHA1程式裡的角色相同。此程式首先顯示一主指示表，其在FOR迴路裡使用一個READ敍述將DATA敍述裡的各選擇功能文字讀入。此為一種有用的技巧，可避免長段的PRINT敍述和具有只需改變DATA敍述裡的資料便可做改變的彈性。在主指示表程式段後，ALOHA1程式裡的全部敍述（從200行起）都放到ALOHA2裡，只是在250行裡改了GOTO的地址。欲重複整個主週期，加上兩個敍述：

```
205 XR=0: IB=0
```

以在畫新的圖景時重新設定XR及IB值。而

```
325 IB=IB+1: ID(IB)=ID: HS(IB)=HS: YR(IB)=YR
```

將打入的ID，HS及YR值存到列陣ID(100)，HS(100)及YR(100)裡，以為往後使用。注意在205行裡IB並未設定為0。

讀寫一文字檔的敍述是在程式的最後一部份。此直接依Apple II磁碟作業系統手册（DOS Manual）的方法所寫成的（參見第8-1節）。

所以在Applesoft要寫一個文字檔，指令

```
OPEN file-name
WRITE file-name
```

送到系統裡，以將此檔打開並寫入資料。在這裡是由Applesoft敍述：

```
D$=CHR$(4)
PRINT D$;"OPEN ";TI$
PRINT D$;"WRITE ";TI$
```

執行的。

由 D$ 代表的 CONTROL-D碼使系統將 PRINT 敍述後的部份解釋爲對系統的指令。在開檔之後，陸續的資料便可用一般 PRINT 敍述將其寫入檔內。在結束時，必須由指令

```
CLOSE file-name
```

將此檔關閉。當然此指令也需用到 PRINT 帶CONTROL-D的敍述。讀入一檔的指令是

```
OPEN file-name
READ file-name
```

同樣的在程式裡，這些指令也要放在 PRINT 含CONTROL-D的敍述裡。然後可用一般的 INPUT 敍述陸續將各資料項讀入。在結束時一樣，也要關閉此檔。

要附加資料在檔上，使用指令

```
APPEND file-name
WRITE file-name
```

當然和上面一樣，也要用 PRINT 含CONTROL-D碼的敍述。

述描一圖景所需的變數爲IB(即其含的圖形塊數目)，及這些圖形塊的ID,HS 及 YR 值。第2160行敍述便將此資料寫入檔內。相對的，當讀

一檔時，第3060到3100行輸入第一個圖景的 IB 值，然後輸入相對的 ID, HS 及 YR 值。這些值正如你自己輸入一樣在畫此圖景時需要用到。程式然後做一小延遲，然後對每一圖景重覆上述程序。

第3030行的 ONERR 敍述使程式避開錯誤的輸入，而第3520行的敍述則將檔關閉以防止在錯誤情況下對磁片產生損壞。

建議練習

1. 將本節討論的有關文字檔原理與ALOHA1的各練習相結合。
2. 容許使用者顯示一文字檔中的第 n 個圖景。使程式能(A)要求輸入 n 值及(B)將第 n 個圖景直接顯示在螢幕上而不必一個個顯示。

ALOHA 2程式

```
100  REM *** ALOHA2-SAVES AND RETRIEVES PANORAMAS ***
110  DIM ID(100),HS(100),YR(100)
120  D$=CHR$(4)
130  FOR LO=1 TO 4: READ TI$(LO): NEXT
140  DATA 1. CREATE PANORAMA,2. SAVE PANORAMA,3. READ PANORAMA,4. QUI
150  TEXT : HOME : HTAB 16: PRINT "OPTIONS"
160  VTAB 4: FOR LO=1 TO 4: HTAB 6: PRINT TI$(LO): NEXT LO
170  PRINT : HTAB 10: INPUT "WHICH OPTION? ";OP
180  ON OP GOTO 200,2000,3000,4000
190  GOTO 150
200  HGR : HCOLOR=3
205  XR=0:IB=0
210  HOME : VTAB 21
220  PRINT "1 DISPLAY A BUILDING BLOCK": PRINT "2 ERASE SCREEN"
230  PRINT "3 RETURN TO MAIN MENU"
240  HTAB 6: INPUT "WHICH OPTION? ";IZ
250  ON IZ GOTO 300,200,150
260  GOTO 210
300  REM *** DISPLAY A BUILDING BLOCK"
310  HOME : VTAB 21
```

```
320  INPUT "ID#, HORIZONTAL SPACE, AND VERTICAL POSITION: ";ID,HS,YR
325  IB=IB+1:ID(IB)=ID:HS(IB)=HS:YR(IB)=YR
330  XR=XR+HS
340  ON ID GOSUB 400,500,600,700,800
350  GOTO 210
400  REM *** DISPLAY A SMALL BIRD ***
410  HPLOT XR,YR-1: HPLOT XR+1,YR TO XR+2,YR: HPLOT XR+3,YR-1
420  XR=XR+4: RETURN
500  REM *** DISPLAY A LARGE BIRD ***
510  HPLOT XR,YR-2: HPLOT XR+1,YR-1: HPLOT XR+5,YR-2
520  HPLOT XR+2,YR TO XR+3,YR: HPLOT XR+4,YR-1
530  XR=XR+6: RETURN
600  REM *** DISPLAY THE LEFT SIDE OF AN ISLAND ***
610  HPLOT XR,YR TO XR+2,YR-3 TO XR+7,YR-3
620  XR=XR+8: RETURN
700  REM *** DISPLAY THE RIGHT SIDE OF AN ISLAND ***
710  HPLOT XR,YR-3 TO XR+5,YR-3 TO XR+7,YR
720  XR=XR+8: RETURN
800  REM *** DISPLAY A PALM TREE ***
805  HPLOT XR,YR-3 TO XR+11,YR-3
810  HPLOT XR+4,YR-4 TO XR+7,YR-4
815  HPLOT XR+5,YR-5 TO XR+6,YR-5
820  HPLOT XR+6,YR-6: HPLOT XR+6,YR-7
825  HPLOT XR+2,YR-6: HPLOT XR+2,YR-7
830  HPLOT XR+3,YR-8: HPLOT XR+5,YR-8
835  HPLOT XR+9,YR-8: HPLOT XR+1,YR-9
840  HPLOT XR+4,YR-9 TO XR+6,YR-9: HPLOT XR+8,YR-9
845  HPLOT XR+2,YR-10 TO XR+3,YR-10
850  HPLOT XR+6,YR-10 TO XR+7,YR-10
855  HPLOT XR+10,YR-10: HPLOT XR+4,YR-11 TO XR+6,YR-11
860  HPLOT XR+9,YR-11: HPLOT XR+7,YR-12 TO XR+8,YR-12
870  XR=XR+12: RETURN
880  HPLOT XR+4,YR-9 TO XR+6,YR-9: HPLOT XR+8,YR-9
890  HPLOT XR+2,YR-10 TO XR+3,YR-10
2000 REM *** SAVE A PANORAMA ***
2010 INPUT "NAME OF FILE TO BE SAVED: ";TI$
2020 INPUT "TYPE B TO BEGIN OR A TO APPEND: ";A$
2030 IF LEFT$(A$,1)="B" THEN 2100
```

```
2040  IF LEFT$(A$,1)="A" THEN 2200
2050  GOTO 2020
2100  REM *** BEGIN A NEW FILE ***
2110  PRINT D$;"OPEN";TI$: PRINT D$;"DELETE";TI$
2120  PRINT D$;"OPEN ";TI$
2130  PRINT D$;"WRITE ";TI$
2140  PRINT IB
2150  FOR LO=1 TO IB
2160  PRINT ID(LO): PRINT HS(LO): PRINT YR(LO):
2170  NEXT LO
2180  PRINT D$;"CLOSE ";TI$
2200  REM *** APPEND TO AN OLD FILE ***
2210  PRINT D$;"APPEND ";TI$
2220  PRINT D$;"WRITE ";TI$
2230  PRINT IB
2240  FOR LO=1 TO IB
2250  PRINT ID(LO): PRINT HS(LO): PRINT YR(LO)
2260  NEXT LO
2270  PRINT D$;"CLOSE ";TI$: GOTO 2300
2300  PRINT "FILE HAS BEEN SAVED": FOR LO=1 TO 1000: NEXT LO
2310   IB=0: GOTO 150
3000   REM *** READ A PANORAMA ***
3010   INPUT "NAME OF FILE TO BE READ: ";TI$
3020   XR=0
3030   HGR : HCOLOR=3: ONERR  GOTO 3500
3040   PRINT D$;"OPEN ";TI$
3050   PRINT D$;"READ";TI$
3060   INPUT IB
3070   IF IB=0 THEN 3100
3080   FOR RE=1 TO IB: INPUT ID,HS,YR:XR=XR+HS
3090   ON ID GOSUB 400,500,600,700,800
3100   NEXT RE
3110   FOR DE=1 TO 1000: NEXT DE
3150   HGR :XR=0: GOTO 3060
3500   FOR DE=1 TO 1000: NEXT DE: REM *** PROVIDE A DELAY ***
3510   POKE 216,0: REM *** TURN OFF ONERR GOTO ***
3520   PRINT D$;"CLOSE ";TI$
3530   GOTO 150
4000   END
```

3—3 建立圖形塊

DESIGN1 程式使你能很快的在螢幕上建立圖形塊。其顯示方法是用一個方形列陣來設計（圖 3-6A 及 3-6B ）。此程式首先要求設計圖形塊所需的行列數，亦即其 raster 大小。然後依此二數顯示出正確的行列陣，並在左上角的格中置一點。此點的功用類似光點。每一格含 10 × 10 螢幕點。此程式最宋容許到 300 格；例如 15 × 20 和 20 × 15 的圖形塊都在此限制內。當然只需稍作改變便可放寬此限制。



圖 3-6A

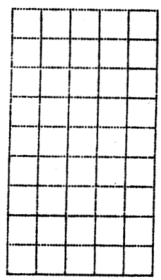


圖 3-6B

程式裡用各種鍵來代替指令：

→	將光點右移一格，如果在某行的尾部則跳到下一行的頭部
←	將光點左移一格，如果在一行的頭部則跳到上一行的尾部
RETURN	將光點下移一格，如在某列的底部則跳到下一列的頂部

U	將光點上移一格，如在某列的頂部則跳到前一列的底部
P	將光點所在的此格包括進此圖形表內並依前面的→鍵移動光點方向
E	將光點所在的格由圖形表中擦去，並依→鍵移動其方向
H	將光點移到左上角
D	顯示出設計的圖形塊中各包括進去的格
C	清除螢幕以繼續設計下一圖形塊
ESCAPE	結束此執行

每當執行 P 時，所在的即格會發亮 。

在輸入 5 × 3 的資料後，按下列各鍵

P→ P→ P→ → P→ P→ P→ P

則螢幕上如圖 3-7 。同樣的結果也可用下列的鍵順序造成

P RETURN P U P RETURN → P → P RETURN P U P

光點在已包括進的格子裡以黑點表示，如在未包括進圖形塊的格子裡則以一亮點表示 。

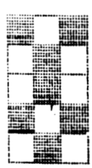


圖 3-7

當你設計完一圖形塊之後，可用類似ALOHA1 及ALOHA2 的程式來利用此圖形塊。下一節裡的DESIGN2 程式便更進步了。其容許使用者〝交談式〞地設計圖形塊，而不需修改任何指令或程式。

在DESIGN1 裡用到的主要變數為：

XR	列數
YR	行數
XA, YA	現行光點座標（對此設計單位而言，非螢幕座標）
XO, YO	前一光點座標（同上）
KP	現行指令（即按鍵）的ASCII 碼
MA$	儲存那一格為亮，那一格為暗的列陣，其格數安排方式是由左到右，由上到下，即先行後列。在列陣中，1 代表此格為亮，0 代表此格為暗

程式首先要求輸入 XR 及 YR 。第 250 到 320 行顯示出此行列陣圖。第 440 行一直做延遲，直到使用者按下一鍵後，將 ASCII 值放到 KP 裡。然後是依按鍵而跳到相對的 450 到 540 行不等，以繼續執行各鍵應有的功能。

建議練習

1. 加入新的控制鍵以容許使用者能以東北，西北，東南，西南的方向移動光點。
2. 將行列陣圖變小以容許處理更大的塊形圖設計，同時也需加大MA$ 列陣。

DESIGN 1程式

```
100  REM *** DESIGN1 FOR BUILDING BLOCKS ***
110  TEXT : HOME : VTAB 21
120  INPUT "NUMBER OF COLUMNS AND ROWS? ";XR,YR
130  DIM MA(XR,YR)
250  REM *** DRAW THE GRID ***
260  HGR : HCOLOR=3
270  FOR Y=10 TO 10*(YR+1) STEP 10
280  HPLOT 10,Y TO 10*(XR+1),Y:
290  NEXT Y
300  FOR X=10 TO 10*(XR+1) STEP 10
310  HPLOT X,10 TO X,10*(YR+1)
320  NEXT X
400  REM *** SET CURSOR AT TOP LEFT HAND POSITION ***
410  XA=1:YA=1:SW=0: GOSUB 700
430  REM *** AWAIT A KEYPUSH ***
440  GET A$:KP=128+ASC(A$):X0=XA:Y0=YA
450  IF KP=136 THEN XA=XA-1: GOTO 630: REM *** LEFT ARROW ***
460  IF KP=149 THEN 610: REM *** RIGHT ARROW ***
470  IF KP=141 THEN YA=YA+1: GOTO 630: REM *** RETURN ***
480  IF KP=213 THEN YA=YA-1: GOTO 630: REM *** U ***
490  IF KP=200 THEN  GOSUB 750: GOTO 410: REM *** H ***
500  IF KP=208 THEN MA(XA,YA)=1: GOSUB 800: GOTO 610: REM *** P ***
510  IF KP=197 THEN MA(XA,YA)=0: HCOLOR=0: GOSUB 800: HCOLOR=3: GOTO
     610:  REM *** E ***
520  IF KP=195 THEN  CLEAR : GOTO 110: REM *** C ***
530  IF KP=196 THEN  GOSUB 900: REM *** D ***
540  IF KP=155 THEN  STOP : REM *** ESC ***
590  GOTO 440
600  REM *** MOVE CURSOR FORWARD ***
610  XA=XA+1
620  REM *** CORRECT FOR MOVEMENT OUT OF ROW OR COLUMN ***
630  IF XA<1 THEN YA=YA-1:XA=XR: GOTO 650
640  IF XA>XR THEN XA=1:YA=YA+1
650  IF YA<1 THEN YA=YR: GOTO 630
660  IF YA>YR THEN YA=1: GOTO 630
670  GOSUB 700: GOTO 430
```

```
700  REM *** PLOT THE CURSOR ***
710  IF MA(XA,YA) THEN  HCOLOR=0
720  HPLOT 10*XA+5,10*YA+5
730  HCOLOR=3
740  REM *** TEST IF DESIGN WAS JUST STARTED ***
750  IF SW=0 THEN SW=1: RETURN
760  REM *** ERASE THE PREVIOUS CURSOR ***
770  HCOLOR=3*(MA(X0,Y0)=1)
780  HPLOT 10*X0+5,10*Y0+5
790  HCOLOR=3: RETURN
800  REM *** PLOT AN ELEMENT ***
810  FOR X=10*XA+1 TO 10*XA+9
820  HPLOT X,10*YA+1 TO X,10*YA+9
830  NEXT X
840  HPLOT 10*XR+XA+20,YA+40
850  RETURN
900  REM *** DISPLAY COORDINATES ***
910  TEXT : HOME
920  FOR Y=1 TO YR: FOR X=1 TO XR
930  IF MA(X,Y) THEN  PRINT X;",";Y
940  NEXT X,Y
950  PRINT : PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: PRINT
960  REM *** BRING HI-RES SCREEN 1 ONTO DISPLAY ***
970  POKE -16304,0: POKE -16297,0
980  RETURN
```

3-4 DESIGN 2程式

DESIGN2 程式首先要求輸入下列選擇功能：

1. 設計一圖形塊（並可用 1 到20範圍內的數做爲 ID 保存起來）。
2. 建立一圖景（圖景當然是許多圖形塊構成的）。
3. 將設計好的圖形塊資料列出，以決定是否要用第 4 點功能重新修正。

4. 容許使用者對已設計之圖形塊重新輸入資料再修正之。

5. 離開此程式。

當使用者打入 1 ，程式表現正如DESIGN1 一樣，除了按ESCAPE鍵可回到主指示表和按 S 鍵可使程式顯示下列訊息：

```
TYPE S TO STORE OR C TO CANCEL:
```

打入C（或任何除了S鍵）鍵使你能繼續設計此圖形塊。按 S 鍵則使程式顯示：

```
NEW ID# (DEFAULT=1):
```

如果還未存任何圖形塊的訊息。

```
NEW ID# (DEFAULT=2):
```

則爲已存入一圖形塊的訊息，等等。按RETURN鍵而不按一數目則使程式使用內定值（ default value ）。如果打入已經用過的 ID 值則會將其代表的及所有較高值的圖形塊破壞掉。在因此而破壞一圖形塊之後，程式會再回來要求做初值的輸入。

欲建立 ALOHA 圖景，首先你要用DESIGN1 所用的技巧建立如圖3-3的五個圖形塊。小鳥大小爲 4 × 2 ，而以圖形塊 1 保存之。大鳥，島嶼左右邊及含棕櫚樹部份的各個圖形塊大小分別爲 6 × 3 ，8 × 4 ，8 × 4 及 12 × 14 ，其保存號依次爲 2 到 5 。

要顯示一圖景，在建立並儲存各圖形塊之後，打入 2 即可。程式然後依ALOHA1 一樣順序的處理，而使用者也應所要求的依次做反應。例如，圖 3-1 的圖景便應以第 2-4 節所述的反應輸入之。

如果你打入 3 ，則顯示出圖形塊的內部代表資料（而非圖形本身）。此代表方法也示範了一般資料結構的技巧。然後程式要求輸入 ID 數。當你打入後，程式便顯示如下：

```
NC%(ID), NR%(ID), NP%(ID), IO%(ID)
XP%(IQ), YP%(IQ) for IQ=IO%(ID)+1 to IO%(ID)+NP%(ID)
```

要打入如此的資料，在主指示表裡打入 4 。然後程式便要求輸入。

DESIGN2 程式用到 DESIGN1 裡所有的變數，另外也加入一些。這裡用南太平洋的島嶼一例來解釋各變數的用法。圖形塊 1 爲圖 3-8A 的小島。程式以下列各項來代表它：

NC%(1)	此圖形塊的列數(4)
NR%(1)	此圖形塊的行數(2)
NP%(1)	亮著的格數(4)
XP%(1)	此圖形塊第一個亮點的 X（列）值(1)
YP%(1)	此圖形塊第一個亮點的 Y（行）值(1)
XP%(2)	此圖形塊第二個亮點的 X 值(2)
YP%(2)	此圖形塊第二個亮點的 Y 值(2)
XP%(3)	此圖形塊第三個亮點的 X 值(3)
YP%(3)	此圖形塊第三個亮點的 Y 值(2)
XP%(4)	此圖形塊第四個亮點的 X 值(4)
YP%(4)	此圖形塊第四個亮點的 Y 值(1)
IO%(1)	在 XP% 及 YP% 列陣中存在於小島前的資料數（0）



圖 3-8A

圖形塊 2 爲圖 3-8B 的大鳥。程式對此的表示法爲

```
NC%(2)=6, NR%(2)=3, NP%(2) =6, IO%(2)=4
XP%(5)=1, YP%(5)=1, XP%(6) =2, YP%(6)=2
XP%(7)=3, YP%(7)=3, XP%(8) =4, YP%(8)=3
XP%(9)=5, YP%(9)=2, XP%(10)=6, YP%(10)=1
```

其中 NC%，NR%，NP% 及 IO% 值分別表示其爲 6 × 3 大小，其中 6 格爲亮點，而在 XP% 和 YP% 列陣中前有四項資料（即小鳥資料）。

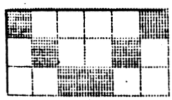


圖 3-8B

圖形塊 3 到 5 分別定義過了。同時程式對其表示法也完全用對圖形塊 1 和 2 的原理一樣。所以各項資料總結各列陣資料爲：

```
ID:   1,  2,  3,  4,  5
NC%:  4,  6,  8,  8, 12
NR%:  2,  3,  4,  4, 14
NP%:  4,  6, 11, 11, 45
IO%:  0,  4, 10, 21, 32
```

此表顯示第一個圖形塊的座標存在 XP%(1)，YP%(1)到 XP%(4)，YP%(4) 裡，而第二個則存在 XP%(5)，YP%(5)到 XP%(10)，YP%(10) 裡。第三個座標存

在XP%(11)，YP%(11)到XP%(21)，YP%(21)裡。第四個則爲XP%(22)，YP%(22)到XP%(32)，YP%(32)，第五個則爲XP%(33)，YP%(33)到XP%(77)，YP%(77)。

此程式所用的主要變數如下：

XR	設計中的圖形塊列數
YR	設計中的圖形塊行數
NA%(500)	用來顯示各圖形塊內何點爲亮與否的標誌（0或1）（參見DESIGN1）
NC%(20)	存入的圖形塊列數
NR%(20)	存入的圖形塊行數
NP%(20)	存入的圖形塊中亮點數
IO%(20)	對 XP% 及 YP% 列陣中此圖形塊前有的資料項
IN	在存某一圖形塊時用來做 XP% 及 YP% 索引
IQ	在顯示圖景中某一圖形塊時用來做 XP% 及 YP% 索引
ID	在儲存圖形塊時各圖形塊的識別號碼
DI	在顯示圖景時各圖形塊的識別號碼
IP	在顯示圖景時，第 DI 個圖形塊存在 XP% 及 YP% 內資料元素共 NP%（DI）個，此變數則爲其中之索引
XX	圖形塊在本身設計行列陣裡的 x 值（非螢幕 x 值）
YY	圖形塊在本身設計行列陣裡的 y 值

DESING1 和 DESIGN2 裡相同的敍述都具有同樣的行數。DESIGN2 的開始部份（即主指示表）具有較多的選擇功能，這些由第 160 到 190 行顯示出。和DESIGN1 相同功能的部份從 250 行到 980 行，其中加入了 550 行和改變了 930 行。在要求使用者輸入存圖形塊的 ID 之後，第1060

行裡用來拿取 MA% 列陣裡連續元素的索引 NO 被設定為 0 ，同時此行也設定了 IO%（ ID ），NR%（ ID ）及 NC%（ ID ）的值。第1070 到1090行的 FOR 迴路檢查 MA% 裡與X ，Y位置（以XX ，YY表示）相對的各元素，每當碰到 1 時， IN 增加 1 ，同時第1080行將 XP%（ IN), YP% (IN) 設定為適當的X ，Y 值。在迴路之後，第 1100行將 NP%（ ID ）設定為座標傳送到XP%，YP% 列陣裡的個數，同時將 ID 加 1 以繼續處理下一個圖形塊。

顯示圖景的部份由1200 行開始。如果輸入為要求顯示一個圖形塊則跳到2090行， HP 值加上由使用者提供的 HS 值。用來做第DI 個圖形塊在 XP%，YP% 列陣裡元素索引的變數 IQ 被設定為 IO%（ DI ），此為第 2100行負責。然後第2110到2130 行的 FOR 迴路將此圖形塊裡的各點逐次加 1 到 NP%（ DI ）為止。（因為此圖形塊共有 NP%（ DI ）個亮點）再依次在第2120 行將各點畫出。迴路之後水平位置 HP 要加上代表圖形塊寬度的 NC%（ DI ），而程式再跳回指示表以容許使用者輸入。

建議練習

1. 用ALOHA1 的練習所述改變或加入更多的選擇功能（ 3-5 ）。
2. 使圖景能存在文字檔中並可重新畫出。
3. 對 DESIGN 1 所提的建議練習做改變。
4. 使各圖形表保存在文字檔內以為往後使用。

DESIGN 2程式

```
100  REM *** DESIGN2 FOR BUILDING BLOCKS AND PANORAMAS ***
110  LOMEM: 16384
120  DIM MA%(300),XP%(500),YP%(500)
130  DIM NR%(20),NC%(20),NP%(20),IO%(20)
140  IN=0:ID=1
```

```
145  REM *** DISPLAY MAIN MENU ***
150  TEXT : HOME
160  PRINT "OPTIONS": PRINT : PRINT "1 TO DESIGN A BUILDING BLOCK"
170  PRINT "2 TO CONSTRUCT A PANORAMA": PRINT "3 TO DISPLAY DATA"
180  PRINT "4 TO TYPE IN A BUILDING BLOCK": PRINT "5 TO QUIT"
190  INPUT "WHICH OPTION? ";ZI
200  ON ZI GOTO 220,2000,3000,4000,5000
210  HOME : GOTO 140
215  REM *** CONSTRUCT A BUILDING BLOCK ***
220  INPUT "NUMBER OF COLUMNS AND ROWS? ";XR,YR
230  N=1: FOR LO=1 TO 300:MA%(LO)=0: NEXT
250  REM *** DRAW THE GRID ***
260  HGR : HCOLOR=3
270  FOR Y=10 TO 10*(YR+1) STEP 10
280  HPLOT 10,Y TO 10*(XR+1),Y
290  NEXT Y
300  FOR X=10 TO 10*(XR+1) STEP 10
310  HPLOT X,10 TO X,10*(YR+1)
320  NEXT X
400  REM *** SET CURSOR AT TOP LEFT HAND POSITION ***
410  XA=1:YA=1:SW=0: GOSUB 700
430  REM *** AWAIT A KEYPUSH ***
440  GET A$:KP=128+ASC(A$):XO=XA:YO=YA
450  IF KP=136 THEN XA=XA-1: GOTO 630: REM *** LEFT ARROW ***
460  IF KP=149 THEN 610: REM *** RIGHT ARROW ***
470  IF KP=141 THEN YA=YA+1: GOTO 630:REM *** RETURN ***
480  IF KP=213 THEN YA=YA-1: GOTO 630: REM *** U ***
490  IF KP=200 THEN  GOSUB 750: GOTO 410: REM *** H ***
500  IF KP=208 THEN GOSUB 5500:MA%(NO)=1: GOSUB 800: GOTO 610:REM *** P ***
510  IF KP=197 THEN HCOLOR=0: GOSUB 5500:MA%(NO)=0: GOSUB 800: HCOLOR=3: REM *E*
520  IF KP=195 THEN 230: REM *** C ***
530  IF KP=196 THEN  GOSUB 900: REM *** D ***
540  IF KP=155 THEN 150: REM *** ESC ***
550  IF KP=211 THEN  GOSUB 1000: REM *** S ***
590  GOTO 430
600  REM *** MOVE CURSOR FORWARD ***
610  XA=XA+1
620  REM *** CORRECT FOR MOVEMENT OUT OF ROW OR COLUMN ***
630  IF XA<1 THEN YA=YA-1:XA=XR: GOTO 650
```

```
 640  IF XA>XR THEN XA=1:YA=YA+1
 650  IF YA<1 THEN YA=YR: GOTO 630
 660  IF YA>YR THEN YA=1: GOTO 630
 670  GOSUB 700: GOTO 430
 700  REM *** PLOT THE CURSOR ***
 710  GOSUB 5500: IF MA%(NO) THEN  HCOLOR=0
 720  HPLOT 10*XA+5,10*YA+5
 730  HCOLOR=3
 740  REM *** ERASE THE PREVIOUS CURSOR IF NECESSARY ***
 750  IF SW=0 THEN SW=1: RETURN
 770  HCOLOR=3*(XO+XR*(YO-1)=1)
 780  HPLOT 10*XO+5,10*YO+5
 790  HCOLOR=3: RETURN
 800  REM *** PLOT AN ELEMENT ***
 810  FOR X=10*XA+1 TO 10*XA+9
 820  HPLOT X,10*YA+1 TO X,10*YA+9
 830  NEXT X
 840  HPLOT 10*XR+XA+20,YA+40
 845  SW=0
 850  RETURN
 900  REM *** DISPLAY COORDINATES ***
 910  TEXT : HOME
 920  FOR YA=1 TO YR: FOR XA=1 TO XR
 930  GOSUB 5500: IF MA%(NO) THEN  PRINT XA;",";YA
 940  NEXT XA,YA
 950  PRINT : PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: PRINT
 960  REM *** BRING HI-RES SCREEN 1 ONTO DISPLAY ***
 970  POKE -16304,0: POKE -16297,0
 980  RETURN
1000  REM *** STORE THE COORDINATES FOR LATER USE ***
1010  HOME : VTAB 21
1020  PRINT "TYPE S TO STORE OR C TO CANCEL: ";: GET A$
1030  IF A$<>"S" THEN  HOME : VTAB 21: GOTO 440
1040  PRINT "NEW ID# (DEFAULT=";ID;"): ";: INPUT ID$
1050  IF LEN(ID$)>0 THEN ID=VAL(ID$)
1060  NO=0:IO%(ID)=IN:NR%(ID)=YR:NC%(ID)=XR
1070  FOR YY=1 TO YR: FOR XX=1 TO XR
1080  NO=NO+1: IF MA%(NO) THEN IN=IN+1:XP%(IN)=XX:YP%(IN)=YY
1090  NEXT XX,YY
1100  NP%(ID)=IN-IO%(ID):ID=ID+1
```

```
1110  HOME : VTAB 21: RETURN
2000  REM *** CONSTRUCT A PANORAMA ***
2010  HP=0: HGR : HCOLOR=3
2020  REM *** DISPLAY SECONDARY MENU ***
2030  HOME : VTAB 21
2040  PRINT "TYPE 1 TO DISPLAY A BUILDING BLOCK"
2050  PRINT "TYPE 2 TO CLEAR THE SCREEN"
2060  INPUT "TYPE 3 TO RETURN TO MAIN MENU: ";Z
2070  ON Z GOTO 2090,2010,150
2080  GOTO 2030
2090  INPUT "ID#, HORIZONTAL SPACING, VERTICAL POSITION";DI,HS,VP
2100  HP=HP+HS:IQ=I0%(DI)
2110  FOR IP=1 TO NP%(DI)
2120  IQ=IQ+1: HPLOT HP+XP%(IQ),VP+YP%(IQ)
2130  NEXT IP:HP=HP+NC%(DI): GOTO 2030
3000  REM *** DISPLAY CODED DATA FOR ANY BUILDING BLOCK ***
3010  REM *** STATEMENTS DO THIS FOR THE CURRENT BLOCK ***
3020  TEXT : HOME :BN=ID: INPUT "ID#? ";ID
3025  PRINT "NUMBER\ OF COLUMNS IS ";NC%(ID)
3030  PRINT "NUMBER OF ROWS IS ";NR%(ID)
3035  PRINT "NUMBER OF POINTS IS ";NP%(ID)
3037  PRINT "BASE ADDRESS IN ARRAY IS ";I0%(ID)
3040  PRINT "X","Y"
3050  POKE 34,5: REM *** LOWERS TOP OF SCROLLING WINDOW ***
3060  FOR L=I0%(ID)+1 TO I0%(ID)+NP%(ID)
3070  PRINT XP%(L),YP%(L)
3080  NEXT L
3090  PRINT : PRINT "PRESS ANY KEY TO CONTINUE";
3100  GET A$:ID=BN: POKE 34,0: GOTO 150
4000  REM *** INPUT CODED BUILDING BLOCK ***
4010  ID=ID+1:I0%(ID)=I
4020  PRINT "ID#(DEFAULT=";ID;"): ";: INPUT ID
4030  INPUT "NUMBER OF COLUMNS AND ROWS? ";NC%(ID),NR%(ID)
4040  INPUT "NUMBER OF POINTS? ";NP%(ID)
4050  IN=I0%(ID)
4060  FOR L=1 TO NP%(ID)
4070  IN=IN+1: INPUT "X AND Y VALUES? ";XP%(IN),YP%(IN)
4080  NEXT L
4090  PRINT "BUILDING BLOCK HAS BEEN STORED"
```

```
4100  FOR DE=1 TO 1000: NEXT DE
4110  GOTO 150
5000  REM *** END ***
5010  END
5500  NO=XA+XR*(YA-1): RETURN
```

3—5 使用Apple圖形表（SHAPE TABLES）

你可繼續在 Apple II 上擴廣類似圖形塊的原理。這些特別的指令使你能夠：

1. 建立內含經過編碼的圖形定義的圖形表。
2. 在螢幕上任何地方畫出這些圖形，並可任意放大縮小或旋轉之。

STATIC 1 程式使用這些特性使你能創造如圖 3-9A 到 3-9F 的圖形。

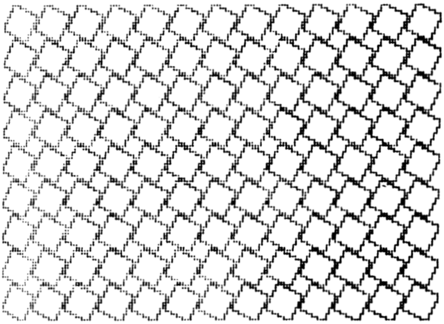


圖　3-9A

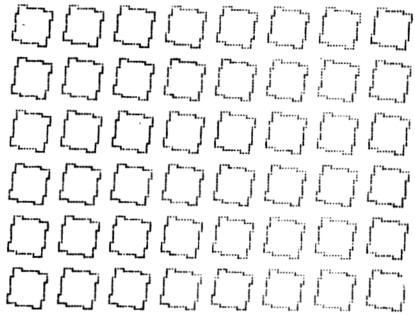


圖　3-9B

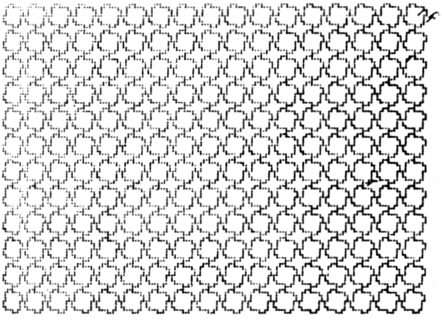


圖　3-9C

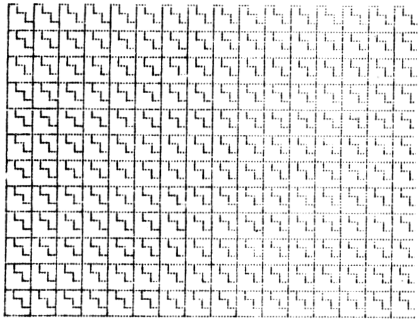


圖　3-9D

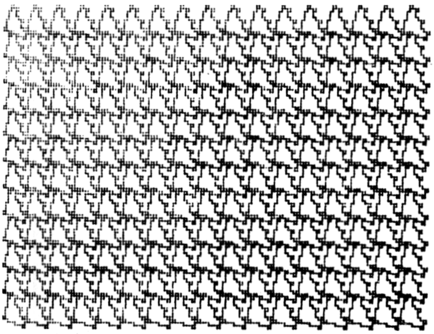


圖　3-9E

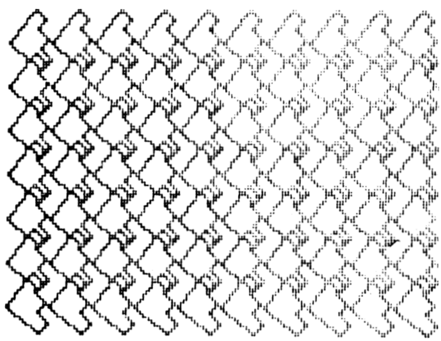


圖　3-9F

圖形 3-9A 到 3-9C的圖形模式由一個存在圖形表中的SHAPE1構成的（參見圖3-10左圖）。而圖 3-9D 到 3-9F 則由圖形表中的 SHAPE2構成的（參見圖 3-10右圖）。



圖 3-10

此程式首先要求輸入(1)圖形號，即 1 或 2 ，(2)大小，此可在 1 到 255 之間及(3)旋轉角度，以 5.625°（即直角的 1 ／ 16 ）爲單位——例如在旋轉部份打入 3 則使圖形旋轉 16.875° 。然後程式將圖形在螢幕上依所指定的大小及旋轉度垂直並水平的連續畫出。

此程式用的變數爲：

NU	圖形號
SI	大小因素
RT	旋轉值（以 5.625° 爲單位）
X , Y	畫圖位置 *x* ， *y* 值
C	用來存圖形表中碼的變數

程式的第一部份，第 120 到 140 行，由 DATA 敍述裡一連串的資料碼讀入並將這些值 POKE 到以7676爲首的連續地址中以建立此圖形表 。（此爲用來建立圖形表的原理，將在後面解釋）一個圖形表內可以含許多不同定義的圖形 。然後此程式要求輸入圖形號，大小及旋轉值 。在範例圖中的

各個輸入值分別爲〔1，4，4〕，〔1，4，2〕，〔1，3，4〕，〔2，4，0〕，〔2，4，4〕及〔2，5，4〕。第 200 行用下列指令設定大小及旋轉值

```
SCALE=SI: ROT=RT
```

然後再用

```
DRAW NU AT X,Y
```

將此圖形畫在X，Y點上，其值皆由重疊 FOR 迴路計算之。

程式然後顯示出

```
PRESS ANY KEY TO CONTINUE
```

當你按下任一鍵之後，程式繼續再重來一遍。

STATIC1 的圖形表的設計法是在繪圖紙上畫出 3 × 3 及 4 × 4 的方格，然後依圖 3-11A 及 3-11B 將圖形繞一圈。

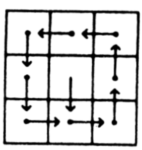


圖　3-11A

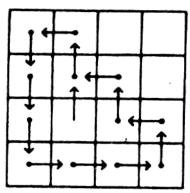


圖　3-11B

如果把它們拉平則結果如圖 3-12所示。（箭尾的一點代表在前進前先畫一點）

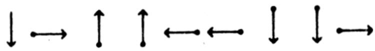


圖 3-12A

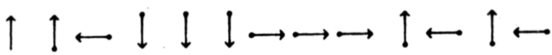


圖 3-12B

在設計類似此類的圖形表時，箭頭可用來代表一次上，右，下，左的移動或代表先畫一點後做上，右，下，左的移動。此共八種不同組合可分別以二進位數 000，001，010，100，101，110 和 111 表示。

此二進制數目然後依由右到左的方式″組裝″（pack）到連續含 8 位元的位元組。最左兩個位元設定爲 0，除非是 001，010或 011 時，此二位元不存放圖形碼，在此三種情形時儲存的形式分別爲 01，10 或 11。圖形碼 000 如果在其左邊都是零時便被忽略而不代表圖形碼，這一點非常重要。最後一個位元組裡全部是 0，即 00 000 000 以告知電腦此處爲圖形組之結尾。所以前面的方形便可編碼或 8 個位元組。下面的表顯示裡面各代表什麼：

1. 二進位制，在各組數之間空格以強調此與圖形之關係。
2. 二進位制中將一位元組以兩個四位元做單位分開。
3. 以十六進位制表示。
4. 以其相對的十進位數表示。

二進位	二進位	十六進位	十進位
00 101 010	0010 1010	2A	42
00 100 100	0010 0100	24	36
00 111 111	0011 1111	3F	63
00 110 110	0011 0110	36	54
00 000 101	0000 0101	05	05
00 000 000	0000 0000	00	00

第二個圖形的相對編碼表爲

二進位	二進位	十六進位	十進位
00 100 000	0010 0000	20	32
00 110 111	0011 0111	37	55
00 110 110	0011 0110	36	54
00 101 101	0010 1101	2D	45
00 100 101	0010 0101	25	37
00 100 111	0010 0111	27	39
00 000 111	0000 0111	07	07
00 000 000	0000 0000	00	00

一個內含 n 個圖形定義的完整圖形表（此例中 $n = 2$ ），內含 $2n+2$ 個元素，後接各個圖形的定義數，即對圖形 1 爲 42，36，…，5，0，對圖形 2 爲 32，55，…，7，0 。

圖形表中的頭兩個數說明此表內含多少圖形，分別以 n 除以 256 的餘數和商表示，此例中即爲 2 和 0 。下一對位元組說明在圖形 1 之定義前有多少個元素，此例爲 6 除以 256 的餘數和商—— 6 和 0 ，次兩個位元組同樣的表示圖形 2 之前的元素個數，此例爲12和 0 。

所以整個圖形表爲

2, 0, 6, 0, 12, 0, 42, 36, 63, 54, 5, 0, 32, 55, 54, 45, 37, 39, 7, 0

此圖形表存在以7676爲首的連續記憶地址中，在一般情況下這樣就很安全了。如果程式很大或有其他特殊情況時便需用到第八章我們討論的記憶位置圖。

建議練習

1. 在STATIC 1 裡建立不同的圖形模式（即仍使用圖形 1 及圖形 2 來建立不同於圖 3-9A 等的模式）。
2. 建立一些不同的圖形並更正 STATIC1 來使用它們。〔提示：改變 DATA 敍述及讀此敍述的 FOR 迴路之上下限〕
3. 寫一程式能要求輸入連續用手編寫的圖形表碼，並將其存入文字檔。
4. 改變ALOHA 1 及 ALOHA 2 以包含我們所定義的圖形。
5. 擴充此程式以容許使用者決定圖形的旋轉及大小。
6. 用第 3-3 和 3-4 節所述的〞交談式〞設計技巧來建立一個設計圖形來的程式。
7. 將第 5 點和第 6 點結合。

STATIC 1程式

```
100  REM *** STATIC1-USES TWO SHAPES ***
110  REM *** STORE THE SHAPE TABLE ***
120  FOR LO=7676 TO 7695: READ C: POKE LO,C: NEXT
130  DATA 2,0,6,0,12,0,42,36,63,54,5,0,32,55,54,45,37,39,7,0
140  POKE 232,252: POKE 233,29: REM *** STORE BASE ADDRESS (7676=29*256+252) ***
150  HGR : HCOLOR=3
160  HOME : VTAB 21
170  INPUT "WHICH SHAPE? ";NU
```

```
180  INPUT "SIZE AND ROTATION? ";SI,RT
190  IF SI>20 THEN SI=20
200  SCALE=SI: ROT=RT
210  FOR Y=20 TO 120 STEP (SI*3)
220  FOR X=50 TO 200 STEP (SI*3)
230  DRAW NU AT X,Y
240  NEXT X,Y
250  PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: PRINT : GOTO 150
```

3—6 STATIC 2 程式

STATIC2 程式使用圖形指令來建立一具輻射對稱的模式（圖3-13）。通過中心的任一條綫將此分爲二個完全相對稱的部份。八條綫的間格都很對稱，所以各方形的中心都在 8 個同心圓上。

每一方格中心的螢幕座標 x ，y 可由簡易的式子代表：

$$x=140+r\cos\theta,\qquad y=95+r\sin\theta$$

此處 r 爲方格中心與螢幕中心的距離，θ 爲此綫與通過螢幕中心的水平行的夾角。在程式的頭部先計算好16種 θ 的 $\sin\theta$ 及 $\cos\theta$ 值並存到列陣裡，此爲節省往後畫圖時的計算時間。

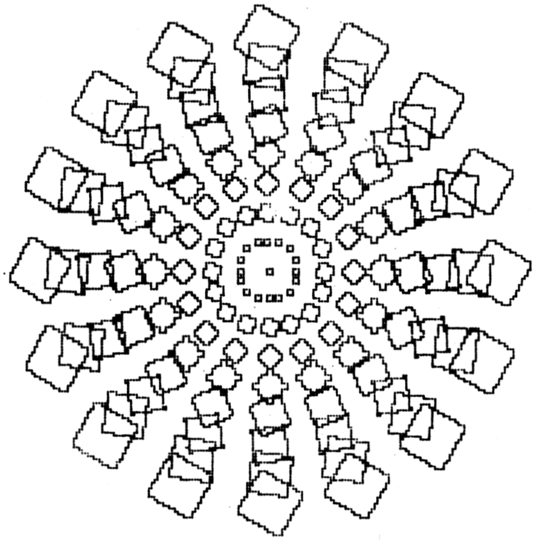


圖 3-13

程式裡主要的變數爲：

TV(1,N)	(N × 2π)／16的餘弦值，用來計算第N條輻射綫方格中心的 *x* 值
TV(2,N)	(N × 2π)／16的正弦值，用來計算相對的 *y* 值
RT	對同一輻射綫上相鄰兩方格的旋轉增量
RO	實際旋轉值
ST	對同一輻射綫上相鄰兩方格的大小增量
SC	實際大小
FT	方格中心與螢幕中心的距離

L1　　計算相鄰方格與中心距離的 FOR 迴路變數

L2　　各輻射綫和 TV 元素的 FOR 迴路變數

此程式先計算後面所需要的正，餘弦值並將其存至 TV 列陣中。接著用一含READ 敍述的 FOR 迴路以將圖形表的數目資料由DATA 敍述中讀入，再存到（ POKE ）適當的記憶位置裡。然後要求輸入某方格的大小及旋轉值。接著用重疊迴路將各方格依適當的大小和角度畫在正確的螢幕位置上。然後再重覆要求輸入方格的大小及旋轉值繼續下去。

建議練習

1. 改變程式顯示的圖形。
2. 容許使用者輸入方格數。
3. 用各種不同的圖形來建立輻射狀圖形。
4. 將螢幕分割成四等分而在每一部份分別做輻射圖形顯示。

STATIC 2程式

```
100  REM *** STATIC2- USES SHAPE TABLES ***
110  REM *** STORE THE TABLE ***
120  FOR LO=7676 TO 7685: READ C: POKE LO,C: NEXT LO
130  DATA 1,0,4,0,58,36,45,54,7,0
140  POKE 232,252: POKE 233,29: REM ***STORE THE BASE ADDRESS OF THE TABLE **
150  REM *** SET UP TABLE OF SINES AND COSINES ***
160  DIM TV(2,16)
170  FOR LO=1 TO 16:TV(1,LO)=COS(TH):TV(2,LO)=SIN(TH):TH=TH+.3925: NEXT LO
180  HOME : VTAB 21: REM *** PROMPT FOR INPUT
190  PRINT "PLEASE ENTER THE AMOUNTS BY WHICH THE "
200  PRINT "ROTATION AND SCALE WILL INCREASE, IN"
210  INPUT "THE RANGES 0 TO 15 AND 0 TO 4: ";RT,ST
220  RO=0:SC=1:FA=10: HGR : HCOLOR=3: SCALE=1: ROT=0
```

```
230  POKE -16302,0: REM *** SWITCH TO FULL SCREEN GRAPHICS ***
240  DRAW 1 AT 140,95
250  FOR L1=1 TO 8
260  FOR L2=1 TO 16
270  DRAW 1 AT 140+TV(1,L2)*FA,95+TV(2,L2)*FA
280  NEXT L2
290  FA=FA+10:SC=SC+ST: SCALE=SC:RO=RO+RT: ROT=RO
300  NEXT L1
310  FOR DE=1 TO 1000: NEXT DE: POKE -16301,0: REM *** MIXED SCREEN GRAPHICS ***
320  GOTO 180
```

3—7 奇怪的方形

SQUARES 程式將螢幕快速的交換塡滿兩種不同的圖形（圖 3-14A 及 3-14B）。其使用的方法是Apple II 的轉換"頁"特性（此將在稍後解釋之）。

第一個模式是用方格對其中心旋轉的簡單圖形指令做出來的。指令

```
HGR2
```

使螢幕顯示到Apple II 高解析度第二頁而在此頁上做畫圖工作，同時不影響第一頁的內容。第二個圖形則爲將方格依其一角爲中心做旋轉。指令

```
POKE -16300, 0
```

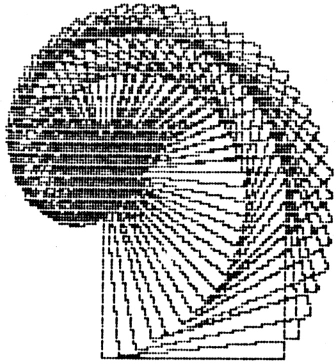


圖　3-14A

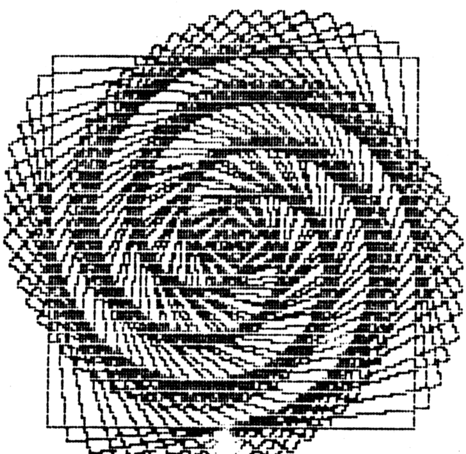


圖　3-14B

將螢幕顯示回到高解析度第一頁，而不影響原保存在第二頁螢幕上的第二個圖模形。指令

```
POKE -16299,0
```

則反轉上述功能。

Apple II 實際上用六頁不同的記憶區來儲存螢幕資料，當執行一適當的指令後便因自動選擇某一塊資料而將其顯示在螢幕上。其中兩頁爲文字模式，兩頁爲低解析度繪圖模式，而最後兩頁爲高解析度繪圖模式。指令 TEXT，GR，HGR 及 HGR2分別選擇螢幕來顯示文字，低解析繪圖，高解析繪圖第一頁及高解析繪圖第二頁。很少會用到文字模式第二頁和低解析繪圖第二頁。

此程式首先用 FOR 迴路將DATA 敍述裡的圖形表資料讀入並儲存起來，同時並放到（POKE）適當的記憶位置裡。圖形表裡的圖形定義一爲以中心爲起點的方形，另一爲以一角爲起點的方形。利用此二圖形配合大小及旋轉值（在 FOR 迴路裡的 SCALE 及 ROT 指令）將圖模形畫出，然後用 POKE 指令來交互轉換兩頁螢幕，同時一個 FOR 迴路會造成延遲。

注意 HGR 及 HGR2 指令不可用在這種快速交換兩頁的顯示中，因爲 HGR 或 HGR2 都分別會自動清除第一頁和第二頁的資料。

建議練習

1. 容許使用者能描述交換頁圖模形中的圖形定義。
2. 修飾 ALOHA2程式以將兩圖景交互顯示。

SQUARES程式

```
100  REM *** SQUARES - USES A SHAPE TABLE ***
110  REM *** STORE THE TABLE ***
120  FOR LO=7676 TO 7690: READ C: POKE LO,C: NEXT
130  DATA 2,0,6,0,9,0,53,39,0,58,36,45,54,7,0
140  POKE 232,252: POKE 233,29: REM *** STORE THE BASE ADDRESS ***
150  HGR : HCOLOR=3: REM *** DRAW ON PAGE 1 ***
160  FOR L=1 TO 64: SCALE=L: ROT=L: DRAW 1 AT 140,80: NEXT L
170  HGR2 : HCOLOR=3: REM *** DRAW ON PAGE 2 ***
180  FOR L=1 TO 64: SCALE=L: ROT=L: DRAW 2 AT 140,80: NEXT
190  POKE -16300,0: FOR DE=1 TO 80: NEXT DE: REM *** DISPLAY PAGE
     1 AND PAUSE ***
200  POKE -16299,0: FOR DE=1 TO 80: NEXT DE: REM *** DISPLAY PAGE
     2 AND PAUSE ***
210  GOTO 190
```

3—8 鐵 鎚

HAMMER 程式使你能用搖捍（如果你有的話）來控制螢幕上鐵鎚的大小及旋轉值。此鐵鎚（圖3-15）由前面討論過的定義圖形表方法定義之。敍述

```
HGR: HCOLOR=3: HPLOT 0,0: CALL 62454
```

將螢幕全部變為白色。一般來說敍，

```
CALL 62454
```

則將螢幕變成最近一次使用的顏色。敍述

```
POKE -16302,0
```

則變成全螢幕繪圖模式，亦即將繪圖區向下擴張而除去原來的四行文字。全螢幕模式時 *y* 的最大值爲 191 。



圖 3-15

敍述

```
ROT=PDL(1)   and   SCALE=1+PDL(2)
```

設定 RO（旋轉值）及 SC（大小值）爲搖捍 1 和 2 的取值（即取決於搖捍位置而在 0 到 255 之間的數值）。敍述

```
XDRAW 1 AT X,Y
```

則在畫圖之前先在原位置上將原圖擦去。鐵鎚的顯示時間由一延遲 FOR 迴路控制。而此程式一直重覆類似的程序。

建議練習

1. 使用兩搖捍來控制一圖形在螢幕上的 *x* ，*y* 值。
2. 使用兩搖捍來控制螢幕上兩圖形的 *x* ，*y* 位置和大小及旋轉值。
3. 使用搖捍來從一大圖形表中選擇一圖形而顯示於螢幕上。

4. 修飾ALOHA 1 及 ALOHA 2 來用搖捍控制各圖形塊的位置。
5. 修飾DESIGN 1 及 DESIGN 2 來用搖捍控制設計圖形塊中將某空格包括在此圖形塊的功能。
6. 將此圖形表及 HAMMER 程式擴充成容許使用者能在螢幕上表現用鐵鎚將一釘子打入木板的動態動作（參見圖 3-16）。

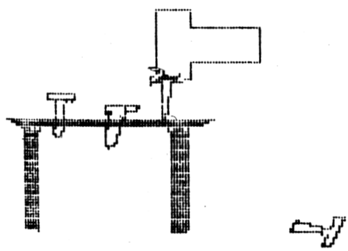


圖 3-16

HAMMER程式

```
100  REM *** HAMMER-USES SHAPES AND PADDLE CONTROL ***
110  REM *** SET UP SHAPE TABLE ***
120  FOR LO=7676 TO 7690: READ PO: POKE LO,PO: NEXT LO
130  DATA 1,0,4,0,54,54,55,46,45,37,60,36,36,63,0
140  POKE 232,252: POKE 233,29: REM *** STORE BASE ADDRESS ***
150  HGR : POKE -16302,0: REM *** FULL SCREEN GRAPHICS ***
160  RO=PDL(1):SC=PDL(0): IF SC=0 THEN SC=1
170  ROT=RO: SCALE=SC
180  XDRAW 1 AT 140,80: REM *** DRAW THE HAMMER ***
190  FOR DE=1 TO 30: NEXT
200  XDRAW 1 AT 140,80: REM *** ERASE THE HAMMER ***
210  GOTO 160
```

第四章　畫　圓

前兩章討論的圖形主要由點和直綫構成的，此章則加入畫圓的能力並顯示圓形圖在各種應用上的有趣用處。想要學習或溫習有關極座標數學基礎的讀者可在此章中看到一些簡單公式推導出解釋，如果你有數學頭腦的話，更能依此而啓發出一些廣泛的幾何結構的做法。

4－1　如何畫圓

無論是在紙上，沙上，或電腦的繪圖螢幕上畫一個圓有很多種方法。在電腦中畫圓精確性和快速性是主要的課題。

由於螢幕是由許多獨立的點構成的（非連續），所以你能得到最好的圓，如果近看的話會有點不協調。圖．1便是電腦能畫出的一個″好″圓。當然如果半徑愈大的話，圓的周圍看起來會愈平滑。

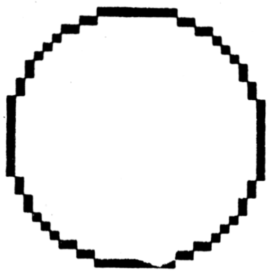


圖 4-1

在電腦繪圖裡，經常用正多邊形來逼近圓的圖形。圖 4-2 便是一系列各邊形而逐漸逼近圓的圖形。在廉價的繪圖螢幕上某些多邊形的邊會因解析度不夠好而呈鋸齒狀如圖 4-3 。

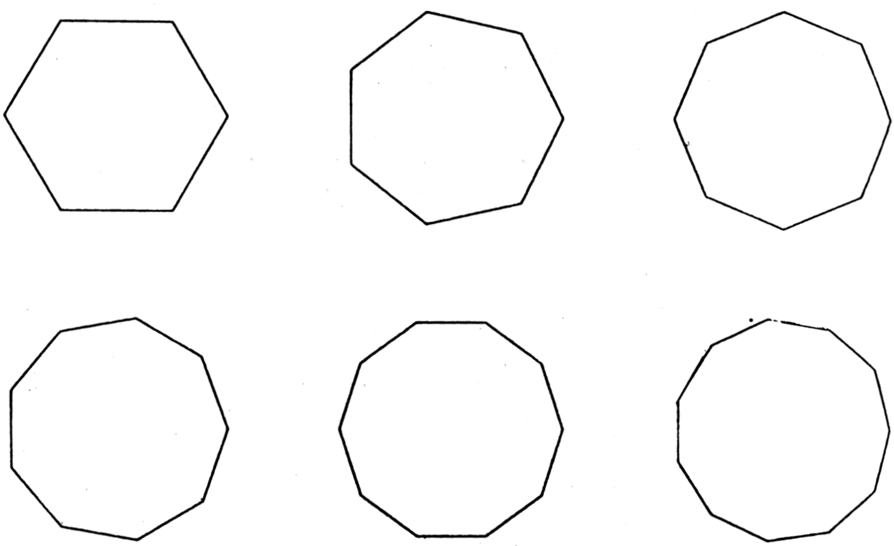


圖 4-2

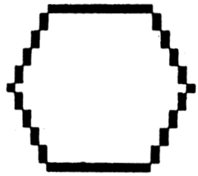


圖 4-3

看了這些近似圓的圖形，我們來看看在 Apple 上所畫的圓。

CIRCLES 程式首先顯示出

```
TYPE THE RADIUS & COORDINATES OF THE CENTER:
```

接著當你打入所要的資料後（以螢幕座標爲單位），此程式便能畫出你要求的圓圖形。其中圓心的 x，y 座標不可小於半徑值，且圓心的 x 值加上半徑不可超過螢幕寬度（279），y 值加上半徑則不可超過螢幕高度（159）。

此程式裡用到的變數爲：

XO , YO	圓心座標
R	圓半徑
X , YU	圓上半部某點之座標
X , YL	圓下半部某點之座標
XP , PU	圓上半部前一點之座標
XP , PL	圓下半部前一點之座

此程式的原理是基於你所熟悉的圓公式

$$(x-x_0)^2+(y-y_0)^2=r^2$$

此公式實際上是由圓上任一點到圓心距離等於半徑所導出。將其展開便可得到相同的公式

$$y^2-2y_0y+(x-x_0)^2+y_0{}^2-r^2=0$$

然後對此式解 y 座標值便得

$$y=y_0\pm\sqrt{(r-x+x_0)(r+x-x_0)}$$

這個式子便是第 160 到 170 行敍述用來對連續的 x 計算相對 y 值的公式。x 值由 X0 − R 逐次加 1 到 X0 + R 爲止。圓上半部的 YU 值取式子裡方根的 + 號，而圓下半部的 YL 值取方根的 − 號。

首先列出此程式的原因是我們對這個圓的數學公式早已熟知。然而在許多應用方面，下兩節所用的畫圓方法却高明的多，其原因我們在討論到這些方法時會詳加解釋，不過 CIRCLES 程式仍然有其作用，例如可做爲學習畫圓的第一步，同時也可用來和別的方法做比較，以了解電腦的特性——多用些數學可節省大量的時間。雖然在大電腦上速度的差異並不明顯，但畢竟你用的是自己買的 Apple 家用電腦！

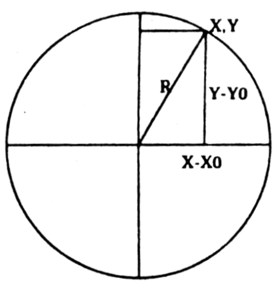


圖 4-4

建議練習

1. 容許使用者說明連續畫點間的 x 值間隔。

2. 對同一 x 值的上下兩 y 值之間畫一直綫以使圓形圖成爲實心圓 。
3. 容許使用者輸入一環形的內外二圓半徑，並使程式依此輸入畫出環形。
4. 使程式畫出一實心環 。
5. 使程式能畫出一圓，內部有數條等分離的垂直綫 。
6. 使程式能畫出一圓，內部有數條等分離的水平行 。〔提示：使FOR 迴路以 y 值做索引來計算相對的兩 x 值〕
7. 畫一圓及其外切正方形 。
8. 畫一圓及其內接正方形 。
9. 改變第 160 行裡求平方根的方法，即用加法來代替乘法以節省計算時間 。〔提示：使用

$$r^2-(x-x_0)^2=(r-x+x_0)(r+x-x_0).$$

調整成此式則只需一次乘法，不過如果

$$s=r^2-(x-x_0)^2, \quad \text{and} \quad s_1=r^2-(x+1-x_0)^2,$$

則

$$s_1-s=2(x_0-x)-1$$

此即爲平方根裡的陳式對每一連續 x 值所增加的量 。利用此式以解決問題〕

11. 輸入兩 x 範圍後畫出在此範圍內的圓的部份圖形 。

CIRCLES程式

```
100  REM *** CIRCLES ***
110  ONERR  GOTO 210: REM *** USED TO BYPASS POINTS OF THE SCREEN ***
120  HGR : HCOLOR=3: HOME : VTAB 21
130  INPUT "TYPE THE RADIUS & COORDINATES OF THE CENTER: ";R,X0,Y0
```

```
140 IF X0-R>=0 AND X0+R<279 AND Y0>=0 AND Y0+R<=159 THEN 16
150 PRINT "PART OF THE CIRCLE WILL BE OFF THE SCREEN"
160 XP=X0-R:PU=Y0:PL=Y0:X=X0-R+1
170 RD=SQR((R-X+X0)*(R+X-X0))
180 YU=Y0+RD:YL=Y0-RD
190 IF YL<0 AND YL>-.01 THEN YL=0
200 HPLOT XP,PU TO X,YU: HPLOT XP,PL TO X,YL
210 XP=X:PU=YU:PL=YL
220 IF X<X0+R THEN X=X+1: GOTO 170
230 PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: PRINT : GOTO 120
```

4—2 小精靈

MR. MOUTH 程式顯示出一實心圓而右邊帶一缺口（參見圖 4-5），如果你有想像力的話，是不是和有名的小精靈很有像呢！

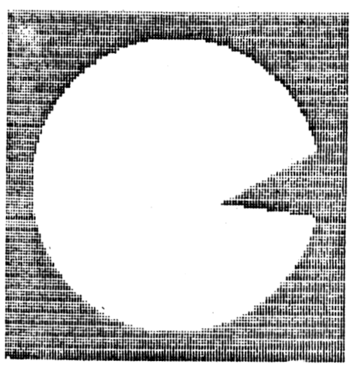


圖 4-5

此程式利用一些簡單的幾何知識——此知識已發展近2000年了。一位早期希臘數學家阿波羅尼斯就是位最早對此種圖形做研究的人。如果我們在圓周上等分數點而將各點與圓心相連，則圓形如圖 4-6 一樣相隔的角度都相等。所以要用來說明圓周上的一點資料有(1)此圓的半徑(2)此點與圓心連綫和正 *x* 軸之夾角。如果用數學符號來表示，則分別爲(1) *r*（代表半徑）及(2) *θ*（代表角度），正如同在 *x*，*y* 平面上描述一點的 *x*，*y* 值一樣。

當走完一圈圓周，則所轉的角度爲360°，如果圓周等分爲10段，以 *x* 軸正向開始，各等分點的值爲 0°，36°，72°，108°，144°…，324°，這些數正好是 360 ／ 10° 的整數倍。Applesoft 裡函數的量數都需以弳度量爲單位，則分別爲 2π／10 的10個整數倍值。要在螢幕上畫一點，需要其 *x*，*y* 座標值。如果以 *r* 和 *θ* 來表示，而圓心座標爲 x_0，y_0，則圓周上一點座標爲

$$x=x_0+r\cos\theta,\qquad y=y_0+r\sin\theta$$

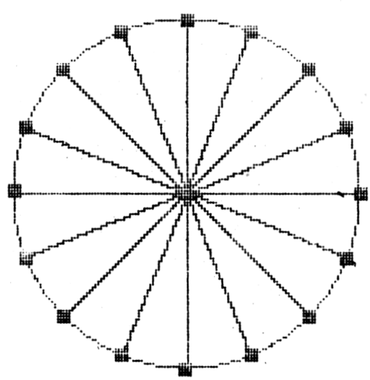


圖 4-6

這些公式實際上是用到最基本的三角函數理論（參見圖 4-7 ）。其中正弦值爲對邊與斜邊的比值，而餘弦值則爲鄰邊與斜邊的比值（圖 4-7 ）。

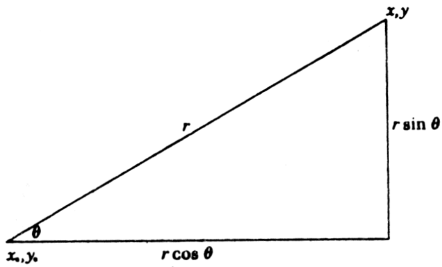


圖 4-7

此程式用到的變數爲：

XO , YO	小精靈的中心
XM , YM	開口兩綫的交點
R	半徑
TC	控制閉嘴時間的因素
TG	控制張開嘴巴時間的因素
SO	控制張嘴最大速度的因素
MS	控制張開嘴巴時最小速度的因素
TO	保持張嘴時間因素
SC	控制閉嘴時最大速度的因素
NO	限制張開嘴巴角度的因素
X%(100)	圓形圖 x 值列陣
Y%(100)	圓形圖 y 值列陣

N	張閉嘴巴的 FOR 迴路索引
K	控制時間延遲的 FOR 迴路索引
TC%	對現行週期相對的 TC 因素
⋮	⋮
NO%	對現行週期相對的 NO 因素

此程式先由DATA 敍述裡取得XC …NO等常數，建立出輪廓的座標，並畫出實心圓。然後開始其主要週期，在週期裡，

1. TC，SO，…，NO 等限制值由隨機的方式計算TC%，SO%，……… NO%。
2. 視 TC% 值保持閉嘴。
3. 視 SO% 值以此速度張嘴。
4. 視 TO% 值保持嘴張開。
5. 視 SC% 值以此速度閉嘴。

程式然後在每週期內重覆取隨機的 TC% 到 NO% 值以張開或閉此嘴巴，直到你打CONTROL-C 。

建議練習

1. 設給此頭部添些裝飾，例如一頂帽子，一隻眼，或一支雪茄。
2. 畫出兩相對的小精靈而交互張閉其嘴部。
3. 使程式能要求使用者輸入以控制動作的各值。
4. 寫一能在圓周上畫出等分數點的程式（如圖 4-8）。
5. 畫出 *n* 段圓弧，其間隔相等，而使用者能輸入(1) *n* 值(2)各圓弧間隔與

圓弧的比率。例如，*n* 爲10時，比例爲 2 比 1 ，則輸出應如圖 4-8 。

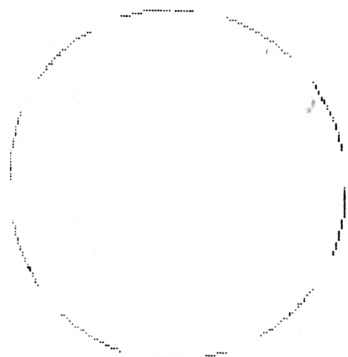


圖 4-8

MR. MOUTH 程式

```
100 REM *** MR. MOUTH ***
110 HOME : VTAB 12: PRINT "PLEASE WAIT ..."
120 DIM X%(314),Y%(314): REM *** ARRAYS FOR X AND Y VALUES ON A CIRCLE
130 READ X0,Y0,R,TC,S0,MS,TG,SC,N0: REM *** PARAMETERS DEFINING THE HE
140 DATA 140,80,50,10,10,10,10,10,35
150 XM=X0+R/3:YM=Y0+R/8: REM *** HINGE OF THE JAWS ***
160 TH=0:DT=.02
170 FOR N=0 TO 314: REM *** LOOP TO  )MPUTE THE X AND Y VALUES ***
180 X%(N)=X0+R*COS(TH):Y%(N)=Y0+R*SIN(TH):TH=TH+DT
190 NEX1 N
200 REM *** SWITCH TO HI-RES GRAPHICS AND TURN THE SCREEN TO WHITE ***
210 HGR   HCOLOR=3: HPLOT 1,1: CALL 62454: HCOLOR=0
220 REM *** DRAW THE SOLID CIRCLE ***
230 FOR N=0 TO 314
240 HPLOT X0,Y0 TO X%(N),Y%(N)
```

```
250  NEXT N
260  REM *** RESET THE RANDOM VARIABLES ***
270  TC%=INT(TC*RND(1)):SO%=INT(SO*RND(1))+MS
280  TG%=INT(TG*RND(1)):SC%=INT(SC*RND(1))+MS
290  NO%=INT(NO*RND(1))
300  REM *** KEEP THE MOUTH CLOSED ***
310  FOR K=1 TO 100*TC%: NEXT K
320  REM *** OPEN THE MOUTH ***
330  HCOLOR=3
340  FOR N=314 TO 314-NO% STEP -1
350  HPLOT XM,YM TO X%(314-N),Y%(314-N)
360  HPLOT XM,YM TO X%(N),Y%(N)
370  FOR K=1 TO 3*SO%: NEXT K
380  NEXT N
390  REM *** KEEP THE MOUTH OPEN ***
400  FOR K=100 TO 3*TG%: NEXT K
410  REM *** CLOSE THE MOUTH ***
420  HCOLOR=0
430  FOR N=314-NO% TO 314
440  HPLOT XM,YM TO X%(314-N),Y%(314-N)
450  HPLOT XM,YM TO X%(N),Y%(N)
460  FOR K=1 TO 3*SC%: NEXT K
470  NEXT N
480  REM *** REPEAT THE CYCLE ***
490  GOTO 260
```

4-3 SON OF MR. MOUTH程式

本節的SON OF MR. MOUTH程式使用簡單的PEEK指令使Apple的喇叭與動態畫面同時發出各種不同的聲響。一旦你RUN此程式，程式便持續其動作，直到你按CONTROL-C鍵爲止。

SON OF MR. MOUTH 程式

```
100  REM *** SON OF MR. MOUTH
110  DIM X%(314),Y%(314): REM *** ARRAYS FOR X AND Y VALUES ON A CIRCLE ***
120  H=-16336: REM ***PEEK ADDRESS TO MAKE A NOISE ***
130  READ X0,Y0,R,TC,SO,MS,TG,SC,NO: REM *** PARAMETERS DEFINING THE HEAD ***
140  DATA 140,80,50,10,10,10,10,10,35
150  TH=0:DT=.02
160  FOR N=0 TO 314: REM *** COMPUTE THE X AND Y VALUES ***
170  X%(N)=X0+R*COS(TH):Y%(N)=Y0+R*SIN(TH):TH=TH+DT
180  NEXT N
190  HOME : VTAB 21: HTAB 16: FLASH : PRINT "WELCOME!": NORMAL
200  REM *** CHANGE THE HI-RES SCREEN TO WHITE ***
210  HGR : HCOLOR=3
220  HPLOT 1,1: CALL 62454: HCOLOR=0
230  REM *** DRAW THE SOLID CIRCLE ***
240  FOR N=0 TO 314
250  HPLOT X0,Y0 TO X%(N),Y%(N)
260  NEXT N
270  REM *** PLOT THE EYE ***
280  HCOLOR=3: FOR MP=153 TO 158: HPLOT MP,55 TO MP,60: NEXT MP
290  HCOLOR=0
300  REM RESET THE RANDOM VARIABLES
310  TC%=INT(TC*RND(1)):SO%=INT(SO*RND(1))+MS
320  TG%=INT(TG*RND(1)):SC%=INT(SC*RND(1))+MS
330  NO%=INT(NO*RND(1))
340  REM *** KEEP THE MOUTH CLOSED ***
350  FOR K=1 TO 100*TC%: NEXT K
360  REM *** OPEN THE MOUTH ***
370  HCOLOR=3
380  FOR N=314 TO 314-NO% STEP -1
390  HPLOT X0,Y0 TO X%(314-N),Y%(314-N)
400  HPLOT X0,Y0 TO X%(N),Y%(N)
410  NN=INT(RND(1)*10)+1
420  FOR K=1 TO SO%: REM *** MAKE RANDOM SOUNDS ***
430  ON NN GOSUB 620,640,660,680,700,720,740,760,760,760
440  NEXT K
450  NEXT N
```

```
460  REM *** KEEP THE MOUTH OPEN ***
470  FOR K=100 TO 3*TG%: NEXT K
480  REM *** CLOSE THE MOUTH ***
490  HCOLOR=0
500  FOR N=314-NO% TO 314
510  HPLOT X0,Y0 TO X%(314-N),Y%(314-N)
520  HPLOT X0,Y0 TO X%(N),Y%(N)
530  NN=INT(RND(1)*10)+1
540  FOR K=1 TO SC%: REM *** MAKE RANDOM SOUNDS ***
550  ON NN GOSUB 620,640,660,680,700,720,740,760,770,770
560  NEXT K
570  NEXT N
580  REM *** REPEAT THE CYCLE ***
590  GOTO 300
600  END
610  REM *** SOUND SUBROUTINES ***
620  RR=PEEK(H)
630  RETURN
640  RR=PEEK(H)+PEEK(H)
650  RETURN
660  RR=PEEK(H)+PEEK(H)+PEEK(H)
670  RETURN
680  RR=PEEK(H)+PEEK(H)+PEEK(H)+PEEK(H)
690  RETURN
700  RR=PEEK(H):RR=PEEK(H):RR=PEEK(H)+PEEK(H)
710  RETURN
720  RR=PEEK(H)+PEEK(H):RR=PEEK(H)*PEEK(H)-PEEK(H)
730  RETURN
740  RR=PEEK(H)*PEEK(H)-PEEK(H)
750  RETURN
760  RR=PEEK(H)+14*1/14-PEEK(H):RR=PEEK(H)
770  RETURN
```

4－4　方形和其他幾何圖形

我們現在再回到有關圓的畫圖程式。本節的 CIRCLOIDS 程式能畫出如圖 4-9A 到 4-9E 的圖形，這些圖形的不同取決於N值。當 N ＝ 2 時，圖形為一圓（圖4-9A）。當 N ＝ 1 時，圖形是個菱形（亦即方形旋轉 45°，圖 4-9B）。N 愈接近 0 時，圖形愈像一組相交的綫，相反的，N 值愈大，則圖形愈像正方形。

此程式簡單的說就是依下列方程式來畫圖。

$$|x-x_0|^n+|y-y_0|^n=r^n$$

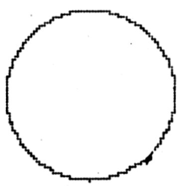


圖 4-9A N=2

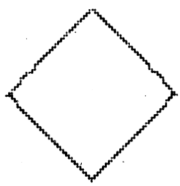


圖 4-9B N=1

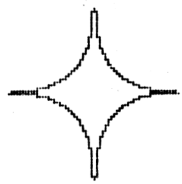


圖 4-9C N=0.25

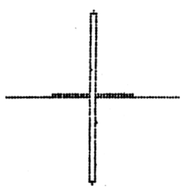


圖 4-9D N=0.1

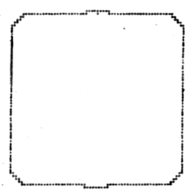


圖 4-9E N=10

如果 $n=2$ 時，此式即爲一圓方程式。如果你有數學基礎的話，的確可自己驗證當 n 很小時趨近成兩條相交的直線，而 n 很大時則趨近一正方形。

程式裡用到變數爲：

N	方程式的 n 值
R	方程式裡的〞半徑〞
XO，YO	中心的 x，y 座標（x_0，y_0）
X，Y	相對於中心的圖形某點座標
XP，YP	相對於中心用來描點的前一 x，y 座標

對在 1 到 R 的 X 值，程式用方程式計算相對 Y 值，然後與前一點座標相連而畫一直線。其他象限的圖形只需依平面的知識改變 X 或 Y 的正負號即可（各圖形皆對原點對稱）。

建議練習

1. 容許使用者在畫下一圖形時保存或清除前面所畫的圖。
2. 使程式能將圖形中封閉區域以實線塡滿。
3. 用程式的對稱性計算並畫出圖形八個對稱點。
4. 如你有數學基礎的話，先對此方程式改成極座標式，然後再用在程式裡。

CIRCLOIDS 程式

```
100  REM *** CIRCLOIDS ***
110  X0=140:Y0=80
120  HGR : HCOLOR=3
130  HOME : VTAB 21
```

```
140  INPUT "RADIUS AND POWER? ";R,N
150  XP=0:YP=R
160  FOR X=1 TO R
170  Y=ABS(X ^ N-R ^ N) ^ (1/N)
180  HPLOT X0-X,Y0+Y TO X0-XP,Y0+YP
190  HPLOT X0-X,Y0-Y TO X0-XP,Y0-YP
200  HPLOT X0+X,Y0+Y TO X0+XP,Y0+YP
210  HPLOT X0+X,Y0-Y TO X0+XP,Y0-YP
220  XP=X:YP=Y
230  NEXT X
240  PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: PRINT : GOTO 130
```

4—5 心臟、腎臟和其他圖形

本節ENVELOPES程式畫出如圖 4-10A 到 4-10C 的圖形。對每一圖形的畫法是在由一特殊方程式決定的輪廓內由一組圓填滿。圖中三個特殊方程式的數學名詞分別爲心臟綫，腎臟綫及 limacon 。

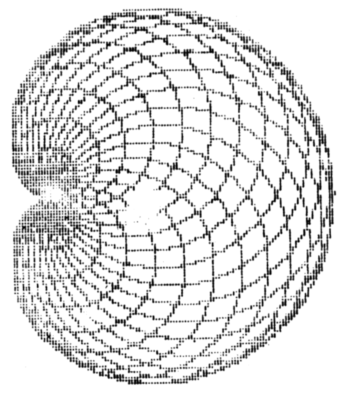


圖 4-10A

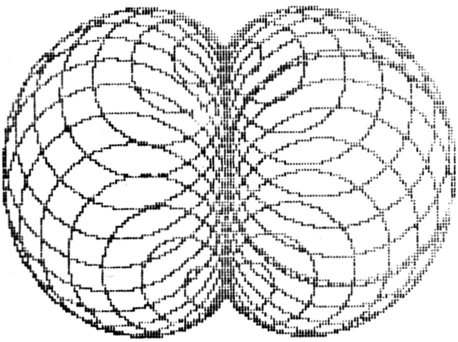


圖 4-10B

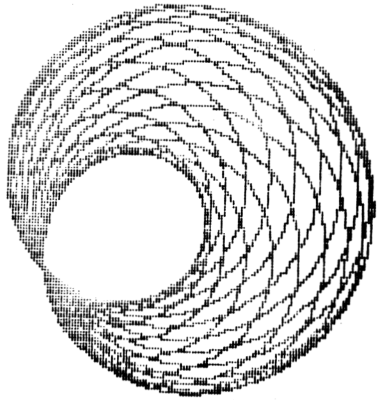


圖 4-10C

此程式首先以（ 80，140 ）點爲圓心，半徑爲25畫一圓，然後在圓（稱爲基準圓）上選一點Q做爲另一通過P點的圓心（參見圖4-11）。接著再在基準圓上選一點做爲通過P點的圓的圓心（參見圖4-12）。此程式連續畫出一系列圓心在基準圓上而都通過P點的圓，同時各圓心在基準圓周上的間隔相等，P點稱之爲cusp點。此圖形最後結果即爲圖 4-10A 。

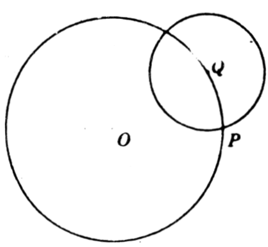


圖 4-11

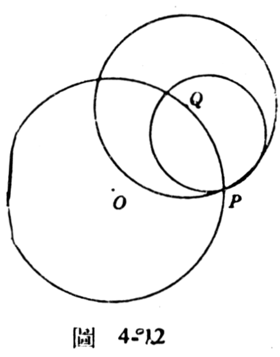


圖 4-12

第二組圓系，其封閉曲綫爲腎臟綫，利用一個基準圓及其垂直直徑（圖4-13）。此程式先在基準圓上選一點做爲要畫的另一個圓的圓心，同時此圓與基準圓的直徑相切（圖4-14）。程式然後再在基準圓上排出一點重覆上述步驟（圖4-15）。

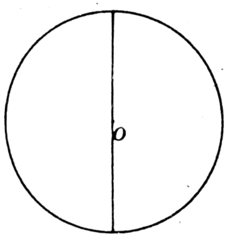


圖 4-13

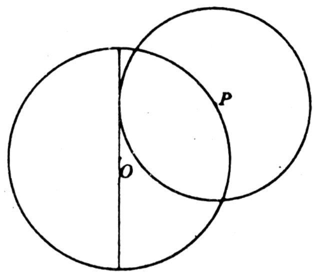


圖 4-14

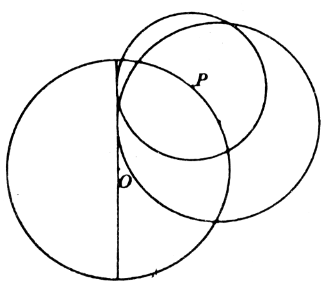


圖 4-15

程式所畫的第三個圖形其封閉曲綫爲數學裡常用的 limacon ，然後使用一個基準圓與其水平直徑（參見圖4-16）。圓系中的每一個圓同樣具有圓心在基準圓上和都過P點的特性（圖4-17），不再詳述。

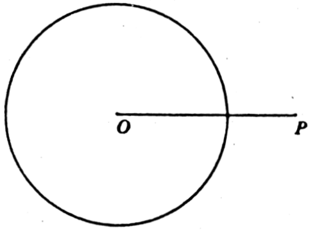


圖 4-16

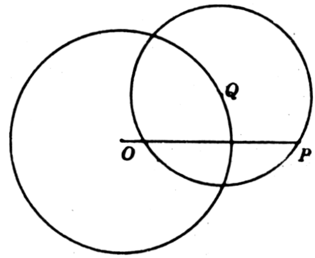


圖 4-17

用來畫這些圓系的公式很簡單，對這三種圖形先在基準圓上畫出以圓心爲原點的兩軸，然後對圓系中各圓取0°到360°的值。令基準圓的半徑爲R（參見圖 4-18A 到 4-18C），則對每一 θ 值所畫出的圓心座標爲

$$x=R\cos\theta,\qquad y=R\sin\theta$$

而半徑爲

$r=2R\cos(\theta/2)$	心臟綫
$r=R\cos\theta$	腎臟綫
$r=\sqrt{(R^2+a^2-2aR\cos\theta)}$	limacon（即圖 4-10C）

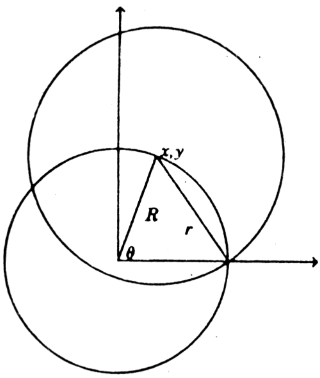


圖 4-18A

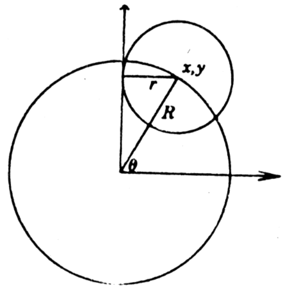


圖 4-18B

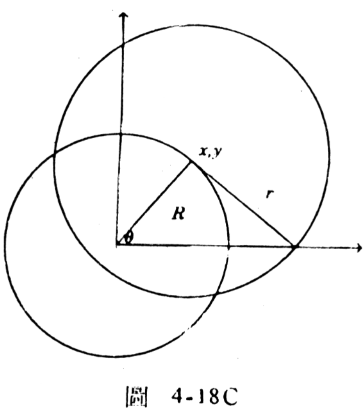


圖 4-18C

ENVELOPES 程式同時也可畫出由直綫構成如圖 4-19 的圖形(astroid)。所有的直綫都等長的由 x 軸畫到 y 軸，其第一象限的截軸座標分別爲（ x ，0 ）及（ 0 ， y ）。若 L 爲此直綫長，則由畢氏定理（圖 4-20 ）可得

$$y = \sqrt{L^2 - x^2}$$

這些綫分別爲 x 値由 0 等分割取到 1 而畫出。在第二，三及四象限裡的截軸座標分別爲〔($-x$, 0), (0 , y)〕,〔($-x$, 0), (0 , $-y$)〕及〔(x , 0), (0 , $-y$)〕。

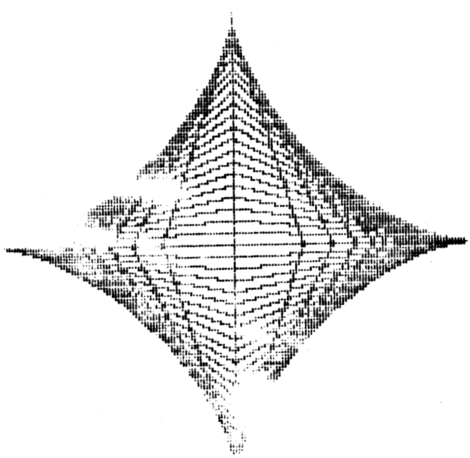


圖 4-19

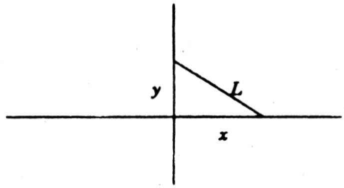


圖 4-20

當程式要求你輸入資料時，如果你不將原圖形清除便畫出第二種圖形，便可建出類似圖4-21的有趣圖形。

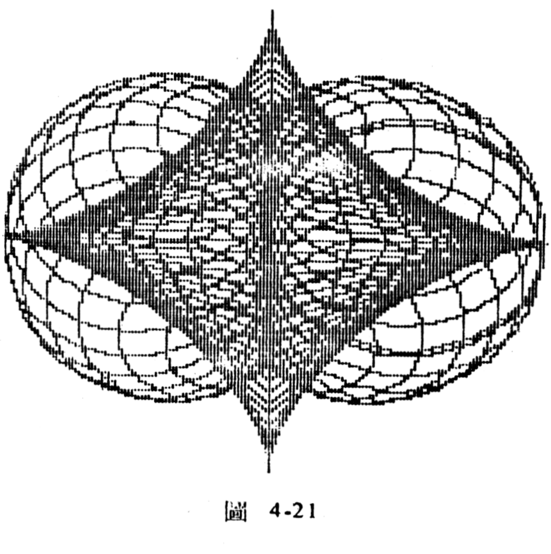


圖 4-21

ENVELOPES程式首先如 MR. MOUTH程式一樣計算出基準圓的等分割18點座標，然後顯示出5種選擇（1到4分別爲畫出圖 4-10A，圖4-10B，圖 4-10C及圖4-19的代碼，5代表"離開此程式"），並要求使用者

選擇一項輸入。接著會問你此圖形是畫在原有圖形上或否。當你選擇 limacon 圖形時（圖 4-10C ），程式會要求輸入螢幕的P點座標（參見圖 4-17）。同時在畫各圖形時，程式也會要求你輸入其解析度（ 1 代表 320 點，2 代表 160 點，3 代表 108 點，等等）。然後程式在 FOR 迴路裡重覆利用畫圓副程式來將適當的圖形畫出，畫完後再回到主指示表。

此程式使用的變數分別為

X (40)	基準圓的 x 座標
NA	X 及 Y 列陣索引
Y (40)	X (40)裡相對的 y 座標
OP	選擇號碼
ST	畫圓時間隔大小——依使用者輸入決定
TH	畫圓系時相對基準圓的夾角，即前面公式裡的 θ
XM , YM	某一圓的圓心座標
RM	上述圓半徑
XC , YC	limacon 圖的 P 的座標，或圖 4-19的中心座標
XN , YN	畫圓系中某圓的點座標
XP , YP	畫圓系中某圓的前一點座標

本程式的整體結構相當直接明瞭，負責各不同圖形部份的程式段，實際上是直接利用前面所述的各種計算公式變成的。

建議練習

1. 容許使用者輸入圓系的圓心及半徑。
2. 容許使用者改變各圖形的P點座標。

3. 容許使用者輸入一角度以改變腎臟綫圖形中垂直直徑爲其他直徑。

4. 容許使用者輸入一角度以旋轉圖4-19的圖形。

ENVELOPES程式

```
100  REM *** ENVELOPES-PRODUCES CARDIOIDS ETC ***
110  TEXT : HOME : VTAB 12: PRINT "<PLEASE WAIT-COMPUTING>"
120  NA=1: DIM X(40),Y(40): REM *** FOR POINTS ON A PROTOTYPE CIRCLE ***
130  FOR A=0 TO .8 STEP .02:X(NA)=COS(A):Y(NA)=SIN(A):NA=NA+1: NEXT A
140  TEXT : HOME : INVERSE
150  VTAB 4:REM *** DISPLAY MAIN MENU
160  HTAB 16: PRINT "OPTIONS": PRINT : PRINT : PRINT
170  FOR LO=1 TO 5: HTAB 6: READ A$: PRINT A$: NEXT : RESTORE
180  NORMAL
190  DATA 1. NEPHROID,2. CARDIOID,3. LIMACON,4. ASTROID,5. QUIT
200  PRINT : PRINT : HTAB 10: INPUT "WHICH OPTION? ";OP
210  HOME : VTAB 21
230  ON OP GOTO 300,400,500,600,680
240  GOTO 130
300  REM *** NEPHROID ***
310  GOSUB 1160
330  INPUT "ENTER THE STEP SIZE? ";ST: IF ST>8 THEN ST=8
340  FOR TH=0 TO 6.28 STEP .195
350  XM=140+40*COS(TH):YM=80+40*SIN(TH):RM=140-XM: GOSUB 1010
360  NEXT TH
370  GOTO 1200
400  REM *** CARDIOID ***
410  GOSUB 1160
430  INPUT "ENTER THE STEP SIZE? ";ST: IF ST>8 THEN ST=8
440  FOR TH=0 TO 6.28 STEP 6.28/27:XM=100+25*COS(TH)
450  YM=80+25*SIN(TH):RM=((XM-75) ^ 2+(YM-80) ^ 2) ^
460  GOSUB 1010
470  NEXT TH
480  GOTO 1200
500  REM *** LIMACON ***
510  GOSUB 1160
530  INPUT "ENTER THE STEP SIZE? ";ST: IF ST>8 THEN ST=8
```

```
540  INPUT "ENTER THE CUSP COORDINATES? ";XC,YC
550  FOR TH=0 TO 6.28 STEP .32:XM=140+15*COS(TH)
560  YM=80+15*SIN(TH):RM=((XM-XC) ^ 2+(YM-YC) ^ 2) ^ .5
570  GOSUB 1010
580  NEXT TH
590  GOTO 1200
600  REM *** ASTROID ***
610  GOSUB 1160
630  INPUT "ENTER LENGTH AND STEP? ";L,ST
640  INPUT "ENTER THE CENTER COORDINATES? ";XC,YC
650  FOR YP=YC-L TO YC+L STEP ST:X=(L ^ 2-(YP-YC) ^ 2) ^ .5
660  HPLOT XC,YP TO XC+X,YC: HPLOT XC,YP TO XC-X,YC: NEXT YP
670  GOTO 1200
680  END
1000  REM *** DRAW A CIRCLE USING 8-FOLD SYMMETRY ***
1010  XP=RM*X(1):YP=RM*Y(1)
1020  NA=2:SW=0
1030  XN=RM*X(NA):YN=RM*Y(NA)
1040  HPLOT XM+XN,YM+YN TO XM+XP,YM+YP
1050  HPLOT XM-XN,YM+YN TO XM-XP,YM+YP
1060  HPLOT XM+XN,YM-YN TO XM+XP,YM-YP
1070  HPLOT XM-XN,YM-YN TO XM-XP,YM-YP
1080  HPLOT XM-YN,YM-XN TO XM-YP,YM-XP
1090  HPLOT XM+YN,YM-XN TO XM+YP,YM-XP
1100  HPLOT XM-YN,YM+XN TO XM-YP,YM+XP
1110  HPLOT XM+YN,YM+XN TO XM+YP,YM+XP
1120  XP=XN:YP=YN
1125  IF SW THEN 1140
1130  NA=NA+ST: IF NA>=40 THEN NA=40:SW=1
1135  GOTO 1030
1140  RETURN
1150  REM *** PROMPT FOR INPUT COMMON TO ALL OPTIONS ***
1160  POKE -16304,0: POKE -16297,0
1170  INPUT "SUPERIMPOSE ON PREVIOUS BACKGROUND? ";SP$
1180  IF LEFT$(SP$,1)="N" THEN  HGR
1190  RETURN
1200  PRINT "HIT ANY KEY TO CONTINUE";: GET A$: PRINT : GOTO 140
```

第五章　統計曲線

5－1　商用圖表

電腦繪圖在商業應用裡所扮演的最重要的角色是幫助你一目了然的了解公司的財務經營狀況，需求及幫助你預測公司的需求和發展前途。利用電腦繪圖的準確度，鮮明的曲綫和動態效果可幫助公司本身的決策人員，顧客或買方做最明智的選擇，同時商用圖表可用在無論地區性或總體性的公司報表和訂單，存貨，財務狀況等各個方面。

商用圖形裡主要的幾種表示法有統計曲綫，長條圖和比率圖等。這些圖形的設計本身就是一種藝術和科學的結合，同時也是許多文章和書籍專論的主題，市面上也有許多專門負責商用圖表的應用軟體。所以看一些簡單的程式和了解其如何畫圖可幫助你選擇這些軟體。

標題和標記在商用圖形裡非常重要，你可藉在圖形螢幕上畫出文字來控制商用圖表的所需要的各種文字標題和說明。DOS TOOL KIT 磁片裡的高解析度繪圖文字產生器便具有這種功能，同時下一節的 FONT FOR-

MER 程式也介紹如何在各種圖形上加入文字的技巧，本章的後面幾節主題則是畫出幾種前面提過的商業圖形表示法。

5−2 建立畫圖用的文字檔

FONT FORMER 程式使你能建立一個要在圖形裡放置文字或符號所需的檔案。要定義一組字元，首先你必須在方格紙上將其畫出（例如以 5 × 7 格的大小），然後找出可用 HPLOT 將此字元在螢幕上畫出的各行頭尾座標，例如字母 A 可將用下列敍述：

```
HPLOT 100,100 TO 100,96 TO 102,94 TO 104,96 TO 104,100
HPLOT 100,98 TO 104,98
```

將A畫在螢幕上（其左上角位在點（100，100））。



圖 5-1

可在 Apple II 繪圖模式裡可用來做常用的文字標記或特殊符號的資料列在此節的最後一部份，每一字元格式爲 5 × 7 點。例如字母 A 便依上述的兩個 HPLOT 敍述分成兩〞段〞編碼。前一〞段〞含五點，正如第一個 HPLOT 敍述畫出五點連綫。第二〞段〞含兩點與第二個 HPLOT 敍述吻合。這些點分別由畫圖紙上 5 × 7 的 A 型字定義得來，x 值爲 0 到 4（由左到右），而 y 值爲 0 到 − 6（由下到上）。字母 A 的整個編碼程序如下：

- 段數：2
- 段1：

  點數：5

  座標：0,0；0,－4；2,－6；4,－4；4,0
- 段2：

  點數：2

  座標：0,－2；4,－2

整個編成的碼按照先爲段數，然後依各段先後順序先排此段點數後接其點座標的順序寫成一串數字，例如對字元A的最後編碼爲

2, 5, 0, 0, 0, −4, 2, −6, 4, −4, 4, 0, 2, 0, −2, 4, −2

當然你也可用建立圖形表的方式來畫出常用的文字及符號，但所花費的功夫（除非你有一個圖形表自動產生器程式）很大，且字元的體積和需要的點數較此法來得多。

一旦定義好需用的字元，便可將其數字串打入 FONT FORMER 程式，此程式也會幫助你檢查定義是否正確。此程式首先顯示出一指示表：

```
1. DEFINE CHARACTERS
2. WRITE A TEXT FILE
3. READ A TEXT FILE
4. DISPLAY A STRING
5. QUIT
   PLEASE TYPE OPTION:
```

然後你打入1，則指示表消失，而螢幕底部出現：

```
NUMBER OF CHARACTERS:
```

輸入你需要定義的字元數後，新的顯示出現：

```
CHARACTER NO. 1
NUMBER OF SEGMENTS:
```

如果你需定義字母A，則打入2，然後螢幕顯示出：

```
CHARACTER NO. 1
SEGMENT NO. 1
NUMBER OF POINTS:
```

依前面我們的定義，打入5，然後程式繼續要求你輸入這5點的座標，接著在螢幕中央將第一段顯示出來，並繼續要求輸入下一段的點數。當整個字元資料輸入完成後，便將整個字元顯示出來，而A一直留在螢幕上直到你按 RETURN（或任一鍵）。然後A消失了，螢幕上繼續顯示要求你輸入第二個字元的定義，此程序一直重覆，直到所有的字元定義完爲止。接著程式再回到主指示表。

現在你可打入2以將整個字元組存到文字檔裡，程式會問你此檔的名稱，寫完檔後再重新回到主指示表。這裡所建立的文字檔可被後面三節裡的程式（或任何你自己設計的程式）重覆使用。要檢查其是否正確，打入3，程式會問你檔案名稱，當你輸入後，程式先清除螢幕，讀入此檔，再將各文字顯示在螢幕上。按任何鍵便可再回到主指示表。

要顯示出一列字串，打入4，程式會要求輸入(1)字串在螢幕上的起始座標，(2)想要依何方向顯示（亦即數字1到4分別代表左到右，右到左，下，上等四種方向），(3)畫字元的方向（亦即數字1到4分別代表字元旋

轉角度爲 0°，90°，180° 及360°，而數字 5 － 8 代表每一字元對其垂直中綫反轉過來或其他型式）及⑷實際字串。對字串 ABCDEFG ，各種顯示方式（相對於要求輸入的第⑶點的數字 1 到 8 ）列在圖 5-2 中。



圖 5-2

欲與 FONT FORMER 及下三節裡程式使用的編碼資料相配位，各字元在文字檔裡的位置應較其在 ASCII 編碼方法裡的位置少 31 。對於用不同的字元或符號在定義時只需對段數輸入打入 0 即可，& 及@就是依次法省略。

本節最後定義的字元資料在螢幕上的結果顯示在圖 5-3 。其相對於 ASCII 碼的對照表如下：

字 元	sp	!	"	#	$	%	&	'	(	)	•	+	,	-		/	0
ASCII	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
字 元	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	@	A
ASCII	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
字 元	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
ASCII	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82
字 元	S	T	U	V	W	X	Y	Z	[	\	]	^	_				
ASCII	83	84	85	86	87	88	89	90	91	92	93	94	95				

```
!"#$% '()*+,-./0123456789:;<=>? ABCDEFGHIJKLM
NOPQRSTUVWXYZ[\]^_
```

圖 5-3

本程式使用的變數如下：

NC	檔案中的字元數
CN	檔案中各順序字元號碼
NS	一字元中的段數
SN	字元中各段的段號，注意對不同的字元並不重新設定爲 1
NP	一段中的點數
PN	字元中各點的點號，注意對不同的段並不重新設定爲 1
PT%(PN,1)	定義一字元的第 PN 個點的 x 值
PT%(PN,2)	定義一字元的第 PN 個點的 y 值
DS%(SN)	在第 SN 段內最後一點的 PN 值
DC%(CN)	在第 CN 個字元中最後一段的 SN 值
XL	螢幕上字元的 x 座標
YL	螢幕上字元的 y 座標
M1	字串寫出的方向碼
M2	字元寫出的方向碼
OP	主指示表的選擇號碼
S1	正在定義、寫或讀的字元的第一段 SN 值
S2	上述字元的最後一段 SN 值

P1	正在定義、寫或讀的段的第一點 PN 值
P2	上述段的最後一點 PN 值
A4$	CONTROL-D字元
FL$	讀或寫的檔名
I	正在顯示的字元在字串中的位置
H%	正常顯示字元時其高度
W%	正常顯示字元時其寬度
H,W	實際畫出的字元高度及寬度

注意此程式所用的資料結構是個很好的範例，請詳細研究之，同時這種方法也用在第 3-4 節的 DESIGN2 程式裡，這種資料結構經常被用在繪圖及其他應用上。

整個程式的運作最好能依其功能順序操作之便能了解。(1)定義文字檔（OP 爲 1），(2)寫入檔（OP 爲 2），(3)讀入檔（OP 爲 3），(4)顯示一字串（OP 爲 4）。各個過程相當直接而易懂。

程式裡用到一些控制螢幕格式的敍述。敍述

```
POKE 34,23
```

對螢幕文字區的後兩行做上移動作（scrolling），此將使用來顯示字元及段數等資料的上兩行保持不動。敍述

```
CALL -958
```

是叫一個將光點後面的螢幕部份清除的機器語言副程式。如果我們不在程式中用它的話，在顯示新的訊息時原有的訊息將保留在螢幕上而將意思弄混。程式裡的各個 HPLOT 的用法都有其簡單的幾何根據。

建議練習

1. 如果你沒有磁碟機，或只需要一小部份的字元組，試將檔案部份的資料用DATA敍述和READ敍述取代，並修改此程式以配合這種情況。
2. 容許使用者除去已設計好的一個字元，並能重新設計它而不影響檔案裡前面的字元。〔提示：在第540後加入除去的功能碼。同時重新設定P1為DS%(S1－1)〕
3. 容許使用者定義字元格式大小。〔提示：在第420行後加入要求輸入W%及H%的敍述。將H及YL值設定為H%，W設定為W%值，並在第380行加入PRINT敍述以將W%及H%寫入磁片〕
4. 設計其他種類的字元檔，例如羅馬字體，或其他字元型狀，並儲存之。
5. 擴充此程式以能使用混合字體，例如不同的大小，不同的字體等。
6. 與第三章的圖形塊原理結合，利用不同的鍵來代表不同的圖形塊。
7. 將本章的原理應用到顯示數學及化學公式的顯示上。
8. 使程式能從不同的字元檔中選出字元而形成一新的字元檔。
9. 修改此程式以能使用圖形定義。

FONT FORMER 程式

```
100  REM *** FONT FORMER ***
110  LOMEM: 16384: REM *** PROTECT MEMORY ***
120  DIM DC%(70),DS%(200),PT%(360,2)
130  W%=6:H%=8:A4$=CHR$(4)
140  TEXT : HOME :H=H%:W=W%: REM *** MAIN MENU ***
150  FOR I=1 TO 5: READ CA$: HTAB 10: PRINT I;".  ";CA$: NEXT I
160  DATA DEFINE CHARACTERS,WRITE A TEXT FILE,READ A TEXT FILE
170  DATA DISPLAY A STRING,QUIT
180  RESTORE
```

```
190  PRINT : INPUT "   PLEASE TYPE OPTION: ";OP
200  ON OP GOTO 230,350,300,590,790
210  PRINT "INVALID OPTION": GOTO 190
220  REM ***SET THE LETTERING MODE ***
230  XL=150:YL=70:M2=1
240  REM *** INITIALIZE SOME PROMPTS ***
250  B1$="NUMBER OF CHARACTERS: "
260  B2$="NUMBER OF SEGMENTS: "
270  B3$="NUMBER OF POINTS: "
280  B4$="COORDINATES: "
290  GOTO 410
300  REM *** SETS THE LETTERING MODE ***
310  XL=0:YL=H:M1=1:M2=1
320  REM *** NULLIFY SOME PROMPTS ***
330  B1$="":B2$="":B3$="":B4$=""
340  REM *** OPEN A FILE ***
350  HOME : INPUT "NAME OF FILE: ";FL$
360  PRINT A4$;"OPEN ";FL$
370  IF OP=3 THEN 390
380  PRINT A4$;"WRITE ";FL$: PRINT NC: GOTO 440
390  PRINT A4$;"READ ";FL$: HGR : HCOLOR=3: GOTO 420
400  REM *** GET THE NO. OF CHARACTERS ***
410  HOME : VTAB 21
420  PRINT B1$;: INPUT "";NC
430  REM *** CYCLE THROUGH THE CHARACTERS ***
440  S2=0:P2=0
450  FOR CN=1 TO NC
460  S1=S2+1
470  ON OP GOTO 480,520,500
480  HGR : HOME : VTAB 21
490  PRINT "CHARACTER NO. ";CN
500  PRINT B2$;: INPUT "";NS
510  S2=S1+NS-1:DC%(CN)=S2: GOTO 530
520  S2=DC%(CN):NS=S2-S1+1: PRINT NS
530  GOSUB 810
540  NEXT CN: REM *** END OF CYCLE ***
550  IF OP=1 THEN 140
560  PRINT A4$;"CLOSE ";FL$
570  GET ZZ$
```

```
580  GOTO 140
590  HGR : HCOLOR=3
600  VTAB 21: CALL -958: INPUT "POSITION OF STRING: ";XL,YL
610  INPUT "DIRECTION OF WRITING: ";M1
620  INPUT "DIRECTION OF LETTER: ";M2
630  POKE 34,23: INPUT "STRING: ";ST$
640  PRINT "STRING: ";ST$;: POKE 34,20
650  IF INT(M2/2)<>M2/2 THEN W=W%:H=H%
660  IF INT(M2/2)=M2/2 THEN H=W%:W=H%
670  IF LEN(ST$)<1 THEN 740
680  FOR I=1 TO LEN(ST$)
690  CN=ASC(MID$(ST$,I,1))-31
700  S1=DC%(CN-1)+1:S2=DC%(CN)
710  P2=DS%(S1-1)
720  NS=S2-S1+1: GOSUB 810
730  NEXT I
740  HOME
750  PRINT "TYPE 1 TO DISPLAY ANOTHER STRING, "
760  INPUT "      2 TO RETURN TO MAIN MENU: ";OQ
770  ON OQ GOTO 600,140
780  GOTO 140
790  END : REM *** QUIT ***
800  REM *** DISPLAY A CHARACTER ***
810  IF NS<1 THEN 1500
820  REM *** CYCLE THROUGH THE SEGMENTS ***
830  FOR SN=S1 TO S2
840  P1=P2+1
850  ON OP GOTO 860,930,890,930
860  HOME : VTAB 21: PRINT "CHARACTER NO. ";CN
870  PRINT "SEGMENT NO. ";SN-S1+1
880  VTAB 23: CALL -958
890  PRINT B3$;: INPUT "";NP
900  P2=P1+NP-1
910  DS%(SN)=P2
920  GOTO 970
930  P2=DS%(SN):NP=P2-P1+1: IF OP=2 THEN  PRINT NP
940  GOTO 970
950  P2=DS%(SN): GOTO 970
960  REM *** CYCLE THROUGH THE POINTS ***
```

```
 970  FOR PN=P1 TO P2
 980  ON OP GOTO 990,1460,1010,1030
 990  PRINT B4$;: INPUT "";PT%(PN,1),PT%(PN,2)
1000  GOTO 1020
1010  INPUT "";PT%(PN,1): INPUT "";PT%(PN,2)
1020  REM *** DISPLAY A SEGMENT ***
1030  ON M2 GOTO 1050,1100,1150,1200,1250,1300,1350,1400
1040  STOP : REM *** INVALID LETTERING MODE ***
1050  IF NP=1 THEN  HPLOT XL+PT%(PN,1),YL+PT%(PN,2)
1060  IF PN<=P1 THEN 1470
1070  OX=PT%(PN-1,1)+XL:OY=YL+PT%(PN-1,2)
1080  HPLOT OX,OY TO PT%(PN,1)+XL,PT%(PN,2)+YL
1090  GOTO 1470
1100  IF NP=1 THEN  HPLOT XL+H+PT%(PN,2),YL-PT%(PN,1)
1110  IF PN<=P1 THEN 1470
1120  OX=XL+H+PT%(PN-1,2):OY=YL-PT%(PN-1,1)
1130  HPLOT OX,OY TO XL+H+PT%(PN,2),YL-PT%(PN,1)
1140  GOTO 1470
1150  IF NP=1 THEN  HPLOT XL+W-PT%(PN,1),YL-H-PT%(PN,2)
1160  IF PN<=P1 THEN 1470
1170  OX=XL+W-PT%(PN-1,1):OY=YL-H-PT%(PN-1,2)
1180  HPLOT OX,OY TO XL+W-PT%(PN,1),YL-H-PT%(PN,2)
1190  GOTO 1470
1200  IF NP=1 THEN  HPLOT XL-PT%(PN,2),YL-W+PT%(PN,1)
1210  IF PN<=P1 THEN 1470
1220  OX=XL-PT%(PN-1,2):OY=YL-W+PT%(PN-1,1)
1230  HPLOT OX,OY TO XL-PT%(PN,2),YL-W+PT%(PN,1)
1240  GOTO 1470
1250  IF NP=1 THEN  HPLOT XL+W-PT%(PN,1)-1,YL+PT%(PN,2)
1260  IF PN<=P1 THEN 1470
1270  OX=XL+W-PT%(PN-1,1)-1:OY=YL+PT%(PN-1,2)
1280  HPLOT OX,OY TO XL+W-PT%(PN,1)-1,YL+PT%(PN,2)
1290  GOTO 1470
1300  IF NP=1 THEN  HPLOT XL+H+PT%(PN,2),YL-W+PT%(PN,1)+1
1310  IF PN<=P1 THEN 1470
1320  OX=XL+H+PT%(PN-1,2):OY=YL-W+PT%(PN-1,1)+1
1330  HPLOT OX,OY TO XL+H+PT%(PN,2),YL-W+PT%(PN,1)+1
1340  GOTO 1470
1350  IF NP=1 THEN  HPLOT XL+PT%(PN,1),YL-H-PT%(PN,2)
```

```
1360  IF PN<=P1 THEN 1470
1370  OX=XL+PT%(PN-1,1):OY=YL-H-PT%(PN-1,2)
1380  HPLOT OX,OY TO XL+PT%(PN,1),YL-H-PT%(PN,2)
1390  GOTO 1470
1400  IF NP=1 THEN  HPLOT XL-PT%(PN,2),YL-PT%(PN,1)
1410  IF PN<=P1 THEN 1470
1420  OX=XL-PT%(PN-1,2):OY=YL-PT%(PN-1,1)
1430  HPLOT OX,OY TO XL-PT%(PN,2),YL-PT%(PN,1)
1440  GOTO 1470
1450  REM *** WRITE COORDINATES TO DISK ***
1460  PRINT PT%(PN,1): PRINT PT%(PN,2)
1470  NEXT PN,SN
1480  ON OP GOTO 1490,1670,1500,1500
1490  GET ZZ$: GOTO 1670
1500  REM *** PROVIDE THE ESCAPEMENT ***
1510  ON M1 GOTO 1550,1580,1610,1640
1520  PRINT "INVALID LETTERING DIRECTION"
1530  PRINT "PRESS ANY KEY AND TRY AGAIN"
1540  GET A$: GOTO 600
1550  IF XL>274-W THEN 1570
1560  XL=XL+W: GOTO 1670
1570  XL=0:YL=YL+H: GOTO 1670
1580  IF XL<W THEN 1600
1590  XL=XL-W: GOTO 1670
1600  XL=280-W:YL=YL+H: GOTO 1670
1610  IF YL>160-H THEN 1630
1620  YL=YL+H: GOTO 1670
1630  YL=0:XL=XL+W: GOTO 1670
1640  IF YL<H THEN 1660
1650  YL=YL-H: GOTO 1670
1660  YL=160-H:XL=XL+W: GOTO 1670
1670  RETURN
1680  REM *** THE END ***
```

簡單字元的編碼

```
sp  0
!   2,1,2,0,2,2,-2,2,-6
"   2,2,1,-6,1,-4,2,3,-6,3,-4
#   4,2,1,0,1,-6,2,3,0,3,-6,2,0,-2,4,-2,2,0,-4,4,-4
```

$	6,2,0,-1,3,-1,1,4,-2,2,1,-3,3,-3,1,0,-4,2,1,-5,3,-5,2,2,-6,2,0
%	5,2,4,-5,0,-1,2,0,-6,1,-6,2,0,-5,1,-5,2,4,0,3,0,2,4,-1,3,-1
&	0
'	1,2,2,-6,2,-4
(	1,4,2,-6,0,-4,0,-2,2,0
)	1,4,2,-6,4,-4,4,-2,2,0
*	3,2,2,0,2,-6,2,0,-5,4,-1,2,0,-1,4,-5
+	2,2,0,-3,4,-3,2,2,-1,2,-5
,	2,1,1,1,2,2,0,2,-1
-	1,2,0,-3,4,-3
.	1,1,3,0
/	1,2,0,-1,4,-5
0	5,2,0,-1,0,-5,2,3,-6,1,-6,2,4,-5,4,-1,2,1,0,3,0,2,3,-5,1,-2
1	2,3,1,-5,2,-6,2,0,2,1,0,3,0
2	5,2,1,-6,3,-6,1,0,-5,2,4,-5,4,-4,2,0,0,4,0,3,0,-1,2,-3,3,-3
3	5,2,0,-6,4,-6,3,4,-5,2,-3,3,-3,2,4,-2,4,-1,2,1,0,3,0,1,0,-1
4	1,5,3 ^ 3,-6,0,-3,0,-2,4,-2
5	4,4,4,-6,0,-6,0,-4,3,-4,2,4,-3,4,-1,2,1,0,3,0,1,0,-1
6	1,5,0,-6,0,0,4,0,4,-3,0,-3
7	1,5,0,-6,4,-6,4,-5,1,-2,1,0
8	7,2,1,0,3,0,2,1,-3,3,-3,2,1,-6,3,-6,2,0,-5,0,-4,2,0,-1,0,-2,2,4,-5,4,-4,2,4,-1,4,-2
9	1,5,4,0,4,-6,0,-6,0,-3,4,-3
:	2,1,2,-2,1,2,-4
;	3,1,2,-3,1,1,1,2,2,0,2,-1
<	1,3,3,-6,0,-3,3,0
=	2,2,0,-2,4,-2,2,0,-4,4,-4
>	1,3,1,-6,4,-3,1,0
?	4,1,2,0,3,2,-2,2,-3,4,-5,2,1,-6,3,-6,1,0,-5
@	0
A	2,5,0,0,0,-4,2,-6,4,-4,4,0,2,0,-2,4,-2
B	4,4,3,0,0,0,0,-6,3,-6,2,0,-3,3,-3,2,4,-1,4,-2,2,4,-4,4,-5
C	5,2,0,-1,0,-5,2,1,-6,3,-6,2,1,0,3,0,1,4,-5,1,4,-1
D	2,4,3,0,0,0,0,-6,3,-6,2,4,-5,4,-1
E	2,4,4,0,0,0,0,-6,4,-6,2,0,-3,3,-3
F	2,3,0,0,0,-6,4,-6,2,0,-3,3,-3
G	5,2,0,-1,0,-5,2,1,0,3,0,2,1,-6,3,-6,1,4,-5,3,4,-1,4,-2,3,-2
H	3,2,0,0,0,-6,2,4,0,4,-6,2,0,-3,4,-3
I	3,2,1,0,3,0,2,1,-6,3,-6,2,2,0,2,-6
J	3,2,4,-6,4,-1,2,1,0,3,0,1,0,-1

```
K   2,2,0,0,0,-6,3,4,-6,1,-3,4,0
L   1,3,0,-6,0,0,4,0
M   1,5,0,0,0,-6,2,-3,4,-6,4,0
N   3,2,0,0,0,-6,2,4,0,4,-6,2,1,-4,3,-2
O   4,2,1,0,3,0,2,1,-6,3,-6,2,0,-1,0,-5,2,4,-1,4,-5
P   3,3,0,0,0,-6,3,-6,2,4,-5,4,-4,2,0,-3,3,-3
Q   5,2,0,-1,0,-5,2,1,-6,3,-6,2,4,-5,4,-1,2,4,0,2,-2,2,1,0,2,0
R   4,3,0,0,0,-6,3,-6,2,4,-5,4,-4,2,3,-3,0,-3,2,2,-3,4,0
S   7,1,0,-1,2,1,0,3,0,2,4,-1,4,-2,2,1,-3,3,-3,2,0,-4,0,-5,2,1,
    -6,3,-6,1,4,-5
T   [illegible],2,[illegible],0,2,-6,2,0,-6,4,-6
U   3,2,0,-6,0,-1,2,4,-1,4,-6,2,1,0,3,0
V   1,5,0,-6,0,-2,2,0,4,-2,4,-6
W   5,2,0,-6,0,0,1,1,-1,2,2,-2,2,-3,1,3,-1,2,4,0,4,-6
X   2,2,0,-6,4,0,2,4,-6,0,0
Y   2,3,2,0,2,-4,0,-6,2,4,-6,3,-4
Z   3,2,0,0,4,0,2,0,-6,4,-6,2,0,-1,4,-5
[   2,4,4,0,0,0,0,-6,4,-6,2,1,0,1,-6
\   1,2,0,-5,4,-1
]   2,4,0,0,4,0,4,-6,0,-6,2,3,-6,3,0
^   1,3,0,-3,2,-5,4,-3
_   1,2,0,0,4,0
```

5—3　簡易曲線程式

本節的各個 PLOT 程式可用來畫含有文字標記，說明，帶在索引軸的各種統計曲綫。其首先暫停一會兒做初設定的工作（實際上是從字元檔中讀入各字元的資料），然後顯示出主指示表：

```
          ***MAIN MENU OPTIONS***
1.    SET DISPLAY OPTIONS
2.    INPUT DATA VALUES
3.    PLOT GRAPH
4.    QUIT
      WHICH OPTION:
```

如果你打入 1 ，接著顯示次指示表，其中帶有各數值的現行值而你可依需要而改變。爲方便起見，各數值都初設定爲其內定值。

```
                OPTIONS
   OPTION                              VALUE
1. SET X-MIN AND X-MAX              -100,100
2. SET Y-MIN AND Y-MAX              -100,100
3. SET AXES                         CROSSED
4. SET TICK MARKS(PLACEMENT)        BOTH AXES
5. SET TICK MARKS(X-AXIS)           10
6. SET TICK MARKS(Y-AXIS)           10
7. GRID                             NULL
8. FRAME                            NULL
9. CAPTION FOR  X-AXIS              X-AXIS
10. CAPTION FOR  Y-AXIS             Y-AXIS
11. TITLE FOR GRAPH                 A GRAPH
12. RETURN TO MAIN MENU
    WHICH\OPTION?
```

如果不需做任何改變，打入12便可直接回到主指示表，或者你也可依任何順序改變其中一個或數個值。在打入任選一數之後，程式又顯示出子指示表，而你可依下列方式反應之：

1. SET X-MIN AND X-MAX ：輸入要劃出的最小及最大 x 值。這些都以使用者座標表示之。
2. SET Y-MIN AND Y-MAX ：輸入資料的最小及最大 y 值，y 值向上爲增加。
3. SET AXES: 依下列規則輸入以決定兩軸的位置：(代表相交於位置 $x = 0$ ，$y = 0$ ；P 代表兩軸在螢幕的左邊和底邊；S 代表不使用。

4. SET TICK MARKS（PLACEMENT）：輸入字母X，Y，B或S分別代表在 x 軸，y 軸，兩軸或不使用文字說明此爲 x 軸或 y 軸。
5. SET TICK MARKS（X-AXIS）：輸入 x 軸（以使用者之單位）的間隔數。
6. SET TICK MARKS（Y-AXIS）：輸入 y 軸的相對間隔數。
7. SET GRID：輸入字母D或N來代表在各間隔上顯示或省略格的標記。
8. SET FRAME：輸入字母D或N來代表在畫圖區是否要將整個圖形分格。
9. CAPTION FOR X-AXIS：輸入 x 軸文字標記。
10. CAPTION FOR Y-AXIS：輸入 y 軸文字標記。
11. TITLE FOR GRAPH：輸入圖形名稱。

當你將各種參數設定好後，打入12便可回到主指示表。然後打入2，以將畫圖資料輸入。程式會問你要畫的點數，及其相對的座標。如果你所輸入的 x 和 y 值超出上述第1項或第2項的範圍，則程式會提醒你重新輸入正確的資料。

當你將正確的資料輸入後，程式會再回到主指示表。打入3便可將各點畫出。在延遲一小段時間後（此時正在輸入另一程式），程式會再回到主指示表，此時再打入3，程式便將各軸，文字標記，單位等依前面的設定畫出，然後在你輸入的各連續點間畫出直綫以將整個曲綫描出。

圖形 5-4A 及 5-4B 顯示出兩種輸出區綫，

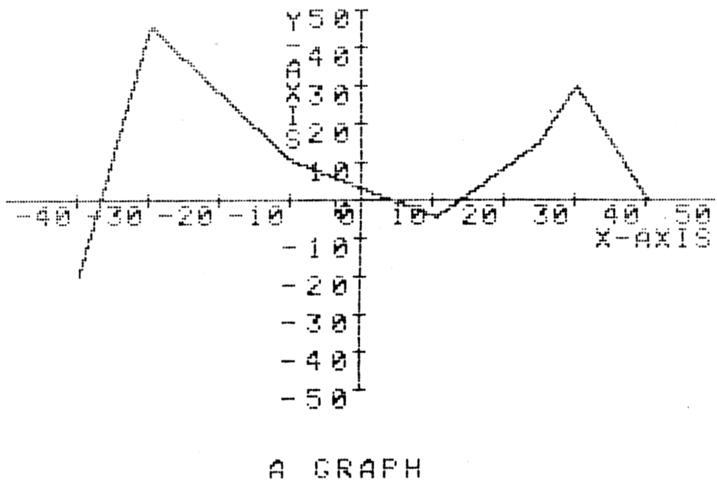

Y-AXIS
50
40
30
20
10
0
-10
-20
-30
-40
-50
X-AXIS
A GRAPH

圖　5-4A

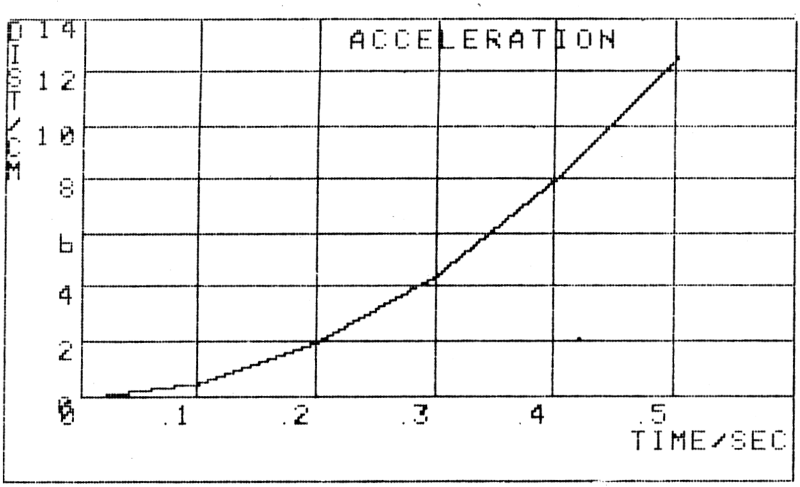

ACCELERATION
DIST/CM
14
12
10
8
6
4
2
0
.1
.2
.3
.4
.5
TIME/SEC

圖　5-4B

此程式用到的主要變數為：

XO , X9	x 值範圍（以使用者座標計）
XO , Y9	y 值範圍（以使用者座標計）
AX$	控制軸位置的選擇
TX , TY	x ， y 軸的分隔距離
TM$	控制分隔位置的選擇
FR$	控制圖形是否需整個分格（如圖 5-4B 即被分格）
XC$, YC$	x ， y 軸的文字標記
TI$	圖形名稱
MM$	主指示表選擇
XL , XR	x 值的螢幕範圍
YL , YR	y 值的螢幕範圍
PT%（300 , 2）	定義文字資料的座標，在5-2 節已解釋
DS%（200）	PT% 列陣的Directory （參見 5-2 節）
DC%（70）	DS% 列陣的Directory （參見 5-2 節）
NC , NS , NP	輸入標記文字的 FOR 迴路範圍
CN , SN , PN	上述各迴路的索引
OP	選擇主指示表的號數
ND	圖形上描點數
XD , YD	某一描點的座標
XU , XS	x 軸的間隔值
YU , YS	y 軸的間隔值
OX , OY	劃一綫起點的螢幕座標
NX , NY	此綫終點座標

PLOT 程式組實際包含三個程式——PLOT 1，PLOT 2 及 PLOT 3。由於各程式佔據的記憶量過大，所以必須分開而視需要交互輸入記憶體內，此利用磁碟作業系統將資料在各程式間傳送並能自動轉換。存放PLOT程式組的磁片上必須有 CHAIN 程式在內（CHAIN 程式在 Apple公司的 system master磁片上），其傳送運接各程式的方法請參閱 Apple磁碟作業系統一書的第八章。

PLOT 1 程式將字元檔內需覆用做文字標記的字元資料讀入。其中各列陣，變數，所使用的邏輯及大部份敍述都與前一節討論的 FONT FOR-MER 程式相同。PLOT 1 程式同時也執行初設定的工作，然後輸入 PLOT 2 程式。PLOT 2 程式顯示主指示表，並處理其中的第 1，2 項功能。當你選擇主指示表的第 3 項功能時（即畫圖），PLOT 2 程式便由 PLOT 3 程式所取代。此程式重新顯示主指示表，故你需再打入 3，才能將螢幕清除並畫出圖形。此圖形一直待在螢幕上直到你按 RETURN（或任一鍵）便回到主指示表，要畫一個新圖而不改變其他資料（即 x 軸設定……等），只需打入 2，程式便要求輸入點數及相對座標，此時只需按一次 3 便可將圖畫出，此因 PLOT3 裡有部份包括了 PLOT2 處理輸入資料的敍述。不過如果你打入的是 1，則 PLOT3 便需由 PLOT2 取代。

因爲 Apple II 記憶量的限制，此圖形是以 HGR 2 模式畫出的，此因程式的某些變數已佔用了 HGR 記憶區之故。

建議練習

1. 容許使用者獨立描出各點或將各點連起來。
2. 容許各資料點標以特別符號。
3. 改變曲線的形式，例如虛綫，以點構成的實綫，或點和綫段交替使用構成的曲綫。

4. 使 y 值向上或向下增大。
5. 使曲綫和一軸間的區域以斜綫填滿。
6. 容許同一圖形上可重覆畫出多條曲綫。
7. 使用者可說明畫圖區的高，寬及位置而不必佔滿整個螢幕。
8. 容許在螢幕上重覆畫曲綫。

PLOT 1 程式

```
100   REM *** PLOT1-FOR MAKING GRAPHS ***
110   HIMEM: 16384
120   DIM TI$(12)
130   HGR2 : TEXT
140   FOR LO=1 TO 12: READ TI$(LO): NEXT LO
150   DATA SET X-MIN AND X-MAX,SET Y-MIN AND Y-MAX,SET AXES
160   DATA SET TICK MARKS(PLACEMENT),SET TICK MARKS(X-AXIS)
170   DATA SET TICK MARKS(Y-AXIS)
180   DATA GRID, FRAME,CAPTION FOR X-AXIS,CAPTION FOR Y-AXIS,
      TITLE FOR GRAPH
190   DATA RETURN TO MAIN MENU
200   X0=-50:X9=50:Y0=-50:Y9=50
210   AX$="CROSSED":TM=10:TM$="BOTH AXES":TY=10
220   ID$="NULL":FR$="NULL":XC$="X-AXIS"
230   YC$="Y-AXIS":TI$="A GRAPH"
240   DIM XD(20),YD(20)
250   FOR LO=1 TO 4: READ MM$(LO): NEXT
260   DATA SET DISPLAY OPTIONS,INPUT DATA VALUES,PLOT GRAPH,QUIT
270   XL=0:XR=279:YU=0:YL=159
280   TEXT : HOME : PRINT "READING IN LABEL DATA"
290   GOSUB 1000
300   PRINT CHR$(4);"BLOAD CHAIN, A520": REM *** CALL PLOT2 ***
310   CALL 520"PLOT2"
1000  REM *** READ IN DATA FOR LABELS FROM A TEXT FILE ***
1010  DIM DC%(70),DS%(200),PT%(360,2)
1020  PRINT CHR$(4);"OPEN LABELTEXT"
1030  PRINT CHR$(4);"READ LABELTEXT"
```

```
1040  INPUT NC: REM *** READS NO. OF CHARACTERS ***
1050  DC%(0)=0:DS%(0)=0
1060  FOR CN=1 TO NC
1070  INPUT NS:DC%(CN)=DC%(CN-1)+NS
1080  IF NS=0 THEN 1140
1090  FOR SN=DC%(CN-1)+1 TO DC%(CN)
1100  INPUT NP:DS%(SN)=DS%(SN-1)+NP
1110  FOR PN=DS%(SN-1)+1 TO DS%(SN)
1120  INPUT PT%(PN,1): INPUT PT%(PN,2)
1130  NEXT PN,SN
1140  NEXT CN
1150  PRINT CHR$(4);"CLOSE LABELTEXT"
1160  RETURN
```

PLOT 2程式

```
100   REM *** PLOT2 ***LIST
500   REM *** MAIN MENU ***
510   TEXT : HOME : INVERSE : HTAB 12: PRINT "***MAIN MENU OPTIONS***"
520   NORMAL
530   VTAB 4: FOR LO=1 TO 4: HTAB 6: PRINT LO;".  ";MM$(LO): NEXT LOP
540   PRINT : PRINT : PRINT
550   HTAB 10: INPUT "WHICH OPTION";OP
560   ON OP GOTO 1000,2000,3000,580
570   VTAB PEEK(37)-1: GOTO 550
580   END
1000  HCOLOR=0: HPLOT 0,0: CALL 62454
1010  TEXT : HOME : INVERSE : HTAB 16: PRINT "OPTIONS"
1020  VTAB 3: HTAB 4: PRINT "OPTION";: HTAB 34: PRINT "VALUE"
1030  VTAB 6: NORMAL
1040  FOR LO=1 TO 11: PRINT LO;".  ";TI$(LO): NEXT LO
1050  POKE 34,18: INVERSE
1060  VTAB 6: HTAB 31: PRINT X0;",";X9,: HTAB 31: PRINT Y0;",";Y9,
1070  HTAB 31: PRINT AX$,: HTAB 31: PRINT TM$,
1080  HTAB 31: PRINT TM: HTAB 31: PRINT TY
1090  HTAB 31: PRINT ID$: HTAB 31: PRINT FR$: HTAB 31
1100  PRINT XC$: HTAB 31: PRINT YC$: HTAB 31: PRINT LEFT$(TI$,10)
1110  PRINT "12.  ";TI$(12)
1120  HOME : PRINT : PRINT : PRINT
```

```
1130  NORMAL : HTAB 5
1140  VTAB 20: INPUT "WHICH OPTION? ";OP%
1150  HOME : PRINT
1160  ON OP% GOTO 1180,1220,1290,1420,1390,1260,1530,1600,1670,
       1710,1750,500
1170  PRINT "NON-EXISTENT OPTION": GOTO 1130
1180  REM *** RESET X MAXIMUM AND X-MINIMUM ***
1190  INPUT "NEW X-MIN AND X-MAX VALUES ARE ";X0,X9
1200  IF X0>X9 THEN XI=X0:X0=X9:X9=XI
1210  GOTO 1010
1220  REM *** RESET Y MAXIMUM AND Y MINIMUM ***
1230  INPUT "NEW Y-MIN AND Y-MAX VALUES ARE ";Y0,Y9
1240  IF Y0>Y9 THEN YI=Y0:Y0=Y9:Y9=YI
1250  GOTO 1010
1260  REM *** SET Y-AXIS TICK INTERVAL ***
1270  INPUT "NEW Y-AXIS TICKMARK INTERVAL IS ";TY:TY=ABS(TY)
1280  GOTO 1010
1290  REM *** SET AXES ***
1300  PRINT "YOU MAY EITHER :"
1310  PRINT "(S)UPPRESS THE AXES": PRINT "(C)ENTER THEM ON THE SCREEN,
1320  PRINT "(P)LACE THEM AT THE BOTTOM LEFT"
1330  INPUT "WHICH ONE? ";IN$
1340  IF LEFT$(IN$,1)="S" THEN AX$="SUPPRESSED": GOTO 1380
1350  IF LEFT$(IN$,1)="C" THEN AX$="CROSSED": GOTO 1380
1360  IF LEFT$(IN$,1)="P" THEN AX$="BOTTOM": GOTO 1380
1370  PRINT CHR$(7): GOTO 1290
1380  GOTO 1010
1390  REM *** SET TICK MARK INTERVAL ***
1400  INPUT "NEW X-AXIS TICK MARK INTERVAL IS ";TM:TM=ABS(TM)
1410  GOTO 1010
1420  REM SET TICK MARK PLACEMENT
1430  PRINT "YOU MAY DISPLAY TICKS ON EITHER: "
1440  PRINT "(X)-AXIS ONLY": PRINT "(Y)-AXIS ONLY": PRINT "(B)OTH AXES"
1450  PRINT "(S)UPPRESSED"
1460  INPUT "WHICH OPTION? ";IN$
1470  IF LEFT$(IN$,1)="X" THEN TM$="X-AXIS": GOTO 1520
1480  IF LEFT$(IN$,1)="Y" THEN TM$="Y-AXIS": GOTO 1520
1490  IF LEFT$(IN$,1)="B" THEN TM$="BOTH AXES": GOTO 1520
1500  IF LEFT$(IN$,1)="S" THEN TM$="SUPPRESSED": GOTO 1520
1510  GOTO 1420
```

```
1520  GOTO 1010
1530  REM *** SET GRID ***
1540  PRINT "GRID IS SET TO ";ID$:
1550  INPUT "NEW GRID SETTING IS(NULL OR DISPLAY)";IN$
1560  IF LEFT$(IN$,1)="N" THEN ID$="NULL": GOTO 1590
1570  IF LEFT$(IN$,1)="D" THEN ID$="DISPLAY": GOTO 1590
1580  GOTO 1540
1590  GOTO 1010
1600  REM *** SET FRAME ***
1610  PRINT "FRAME IS SET TO ";FR$
1620  INPUT "NEW FRAME SETTING IS(NULL OR DISPLAY)";IN$
1630  IF LEFT$(IN$,1)="N" THEN FR$="NULL": GOTO 1660
1640  IF LEFT$(IN$,1)="D" THEN FR$="DISPLAY": GOTO 1660
1650  GOTO 1610
1660  GOTO 1010
1670  REM *** SET CAPTION FOR X AXIS ***
1680  PRINT "CAPTION SHOULD READ: ": HTAB 8: INPUT "";XC$
1690  XC$=LEFT$(XC$,8)
1700  GOTO 1010
1710  REM *** SET Y CAPTION ***
1720  PRINT "CAPTION SHOULD READ: ": HTAB 8: INPUT "";YC$
1730  YC$=LEFT$(YC$,8)
1740  GOTO 1010
1750  REM *** INPUT TITLE FOR GRAPH ***
1760  PRINT "TITLE SHOULD READ": HTAB 10: INPUT "";TI$
1770  TI$=LEFT$(TI$,14)
1780  GOTO 1010
2000  REM *** COLLECT DATA VALUES ***
2010  TEXT : HOME : HTAB 10: INVERSE : PRINT "SUBROUTINE TO COLLECT DATA"
2020  VTAB 4
2030  INPUT "HOW MANY DATA VALUES(0 FOR MAIN MENU)";ND
2040  IF ND>20 THEN  FLASH : GOTO 2130
2050  IF ND=0 THEN 500
2060  NORMAL
2070  FOR IN=1 TO ND
2080  PRINT "PLEASE ENTER DATA PAIR #";IN;: INPUT " ";XD,YD
2090  IF XD<X0 OR YD<Y0 OR XD>X9 OR YD>Y9 THEN 2120
2100  XD(IN)=XD:YD(IN)=YD
```

```
2110  NEXT IN: GOTO 500
2120  FLASH : PRINT "ILLEGAL VALUE": NORMAL : VTAB PEEK(37)-1: GOTO 2080
2130  PRINT "NO MORE THAN 20, PLEASE": INVERSE : GOTO 2030
3000  PRINT CHR$(4);"BLOAD CHAIN, A520"
3010  PRINT "LOADING PLOT3"
3020  CALL 520"PLOT3"
```

PLOT 3程式

```
 100  REM *** PLOT3 ***
 500  REM *** MAIN MENU ***
 510  TEXT : HOME : INVERSE : HTAB 12: PRINT "***MAIN MENU OPTIONS***"
 520  NORMAL
 530  VTAB 4: FOR LO=1 TO 4: HTAB 6: PRINT LO;"   ";MM$(LO): NEXT LOP
 540  PRINT : PRINT : PRINT
 550  HTAB 10: INPUT "WHICH OPTION";OP
 560  ON OP GOTO 1000,2000,3000,580
 570  VTAB PEEK(37)-1: GOTO 550
 580  END
1000  PRINT CHR$(4);"BLOAD CHAIN, A520"
1010  HOME : HTAB 13: PRINT "LOADING PLOT2"
1020  CALL 520"PLOT2"
2000  REM *** COLLECT DATA VALUES ***
2010  TEXT : HOME : HTAB 10: INVERSE : PRINT "SUBROUTINE TO COLLECT DATA"
2020  VTAB 4
2030  INPUT "HOW MANY DATA VALUES(0 FOR MAIN MENU)";ND
2040  IF ND>20 THEN  FLASH : GOTO 2130
2050  IF ND=0 THEN 500
2060  NORMAL
2070  FOR IN=1 TO ND
2080  PRINT "PLEASE ENTER DATA PAIR #";IN;: INPUT " ";XD,YD
2090  IF XD<X0 OR YD<Y0 OR XD>X9 OR YD>Y9 THEN 2120
2100  XD(IN)=XD:YD(IN)=YD
2110  NEXT IN: GOTO 500
```

```
2120 FLASH : PRINT "ILLEGAL VALUE": NORMAL : VTAB PEEK(37)-1: GOTO 2080
2130 PRINT "NO MORE THAN 20, PLEASE": INVERSE : GOTO 2030
3000 REM *** PLOT GRAPHS ***
3010 POKE -16297,0: POKE -16302,0: POKE -16304,0: POKE -16299,0
3020 HCOLOR=3
3030 XS=250/(X9-X0)
3040 YS=130/(Y9-Y0)
3050 IF LEFT$(AX$,1)="S" THEN 3810
3060 IF LEFT$(AX$,1)="C" THEN 3500
3070 IF LEFT$(FR$,1)="N" THEN 3090
3080 HPLOT XL,YU TO XR,YU TO XR,YL TO XL,YL TO XL,YU
3090 HPLOT 29,0 TO 29,130 TO 279,129
3100 IF LEFT$(TM$,1)="S" THEN 3230
3110 IF LEFT$(TM$,1)="Y" THEN 3160
3120 FOR X=29 TO 279 STEP XS*TM
3130 HPLOT X,130 TO X,127:XV=X:YV=139:V=X0+(X-29)/XS
3140 GOSUB 5500: NEXT X
3150 IF LEFT$(TM$,1)="X" THEN 3230
3160 FOR Y=130 TO 0 STEP -YS*TY
3170 HPLOT 30,Y TO 33,Y
3180 V=Y9-Y/YS: IF ABS(V)<.003 THEN V=0
3190 V$=STR$(V):XV=30-9*LEN(V$)
3200 YV=Y+6: FOR LA=1 TO LEN(V$)
3210 C$=MID$(V$,LA,1): GOSUB 5750
3220 XV=XV+9: NEXT LA,Y
3230 GOSUB 5000
3240 GOSUB 4500
3250 IF LEFT$(ID$,1)="N" THEN 3320
3260 FOR X=29 TO 279 STEP XS*TM
3270 HPLOT X,0 TO X,130
3280 NEXT X
3290 FOR Y=130 TO 0 STEP -YS*TY
3300 HPLOT 30,Y TO 279,Y
3310 NEXT Y
3320 XV%=1:YV%=148: GOSUB 5580
3330 GET A$
3340 GOTO 500
3500 REM *** AXES CROSSED ***
3510 C1=YS*Y9:C2=279-XS*X9
3520 HPLOT 29,C1 TO 279,C1
```

```
3530  HPLOT C2,0 TO C2,130
3540  IF LEFT$(TM$,1)="Y" THEN 3670
3550  IF LEFT$(TM$,1)="S" THEN 3810
3560  FOR X=C2 TO 279 STEP XS*TM
3570  HPLOT X,C1-2 TO X,C1+2
3580  YV=C1+8:XV=X:V=X0+(X-29)/XS
3590  GOSUB 5500
3600  NEXT X
3610  FOR X=C2 TO 30 STEP -TM*XS
3620  HPLOT X,C1-2 TO X,C1+2
3630  YV=C1+8:XV=X:V=X0+(X-29)/XS
3640  GOSUB 5500
3650  NEXT X
3660  IF LEFT$(TM$,1)="X" THEN 3810
3670  FOR Y=C1 TO 0 STEP -TY*YS
3680  HPLOT C2-2,Y TO C2+2,Y
3690  V=Y9-Y/YS:V$=STR$(V):XV=C2-9*LEN(V$)
3700  YV=Y+6: FOR LA=1 TO LEN(V$)
3710  C$=MID$(V$,LA,1): GOSUB 5750
3720  XV=XV+9
3730  NEXT LA,Y
3740  FOR Y=C1 TO 130 STEP TY*YS
3750  HPLOT C2-2,Y TO C2+2,Y
3760  V=Y9-Y/YS:V$=STR$(V):XV=C2-9*LEN(V$)
3770  YV=Y+6: FOR LA=1 TO LEN(V$)
3780  C$=MID$(V$,LA,1): GOSUB 5750
3790  XV=XV+9
3800  NEXT LA,Y
3810  IF LEFT$(FR$,1)="N" THEN 3830
3820  HPLOT 279,0 TO 279,130 TO 30,130 TO 30,0 TO 279,0
3830  GOSUB 5000
3840  GOSUB 4500
3850  IF LEFT$(ID$,1)="N" THEN 4010
3860  FOR X=C2 TO 279 STEP TM*XS
3870  HPLOT X,0 TO X,130
3880  NEXT X
3890  FOR X=C2 TO 30 STEP -XS*TM
3900  HPLOT X,0 TO X,130
3910  NEXT X
3920  FOR Y=C1 TO 0 STEP -TY*YS
```

```
3930  HPLOT 30,Y TO 279,Y
3940  NEXT Y
3950  FOR Y=C1 TO 130 STEP TY*YS
3960  HPLOT 30,Y TO 279,Y
3970  NEXT Y
3980  FOR X=C2-2 TO C2+2
3990  HPLOT X,C1-2 TO X,C1+2
4000  NEXT
4010  XV%=C2-26:YV%=C1+16
4020  GOSUB 5580
4030  GET A$: GOTO 500
4500  REM *** DRAW TITLE OF GRAPH ***
4510  XV=115
4520  YV=10
4530  IF LEFT$(AX$,1)="C" THEN YV=160
4540  FOR LO=1 TO LEN(TI$):XV=XV+8:C$=MID$(TI$,LO,1)
4550  GOSUB 5750
4560  NEXT LO
4570  RETURN
4750  END
5000  OX=30+(XD(1)-X0)*XS
5010  OY=-(YD(1)-Y0)*YS+130
5020  FOR NP=2 TO ND
5030  NX=30+(XD(NP)-X0)*XS
5040  NY=130-(YD(NP)-Y0)*YS
5050  HPLOT OX,OY TO NX,NY
5060  OX=NX:OY=NY
5070  NEXT NP
5080  RETURN
5500  V$=STR$(V)
5510  XV=XV-7*LEN(V$)
5520  FOR LA=1 TO LEN(V$)
5530  C$=MID$(V$,LA,1)
5540  GOSUB 5750
5550  XV=XV+7
5560  NEXT LA
5570  RETURN
5580  XV=XV%:YV=7
5590  FOR LA=1 TO LEN(YC$):C$=MID$(YC$,LA,1)
5600  GOSUB 5750
```

```
5610  YV=YV+8
5620  NEXT LA
5630  XV=279-7*LEN(XC$):YV=YV%
5640  FOR LA=1 TO LEN(XC$)
5650  C$=MID$(XC$,LA,1)
5660  GOSUB 5750
5670  XV=XV+7
5680  NEXT LA
5690  RETURN
5750  CN=ASC(C$)-31
5760  S1=DC%(CN-1)+1:S2=DC%(CN)
5770  IF S1>S2 THEN  RETURN
5780  FOR SN=S1 TO S2
5790  P1=DS%(SN-1)+1:P2=DS%(SN)-1
5800  IF P2>=P1 THEN 5830
5810  HPLOT XV+PT%(P1,1),YV+PT%(P1,2)
5820  GOTO 5860
5830  FOR PN=P1 TO P2
5840  HPLOT XV+PT%(PN,1),YV+PT%(PN,2) TO XV+PT%(PN+1,1),YV+PT%(PN+1,2)
5850  NEXT PN
5860  NEXT SN
5870  RETURN
```

5—4 長條圖

長條圖對許多種商業及其他資料具有統計的作用。本節 BAR CHART 程式便能以長條圖的形式統計資料。首先本程式顯示出主指示表，而某些功能值取爲內定值。

```
              *** MAIN MENU ***
         OPTION                      VALUE
      1. DIRECTION OF BARS           VERTICAL
      2. WIDTH OF BARS               15
      3. GAP BETWEEN BARS            10
```

```
4. AXIS DATA                 100,20,5
5. INPUT TITLE
6. INPUT DATA VALUES
7. QUIT
 PLEASE TYPE OPTION:
```

如果你同意這些內定值，並想畫出一個如圖 5-5 的長條圖，只需按 6 鍵即

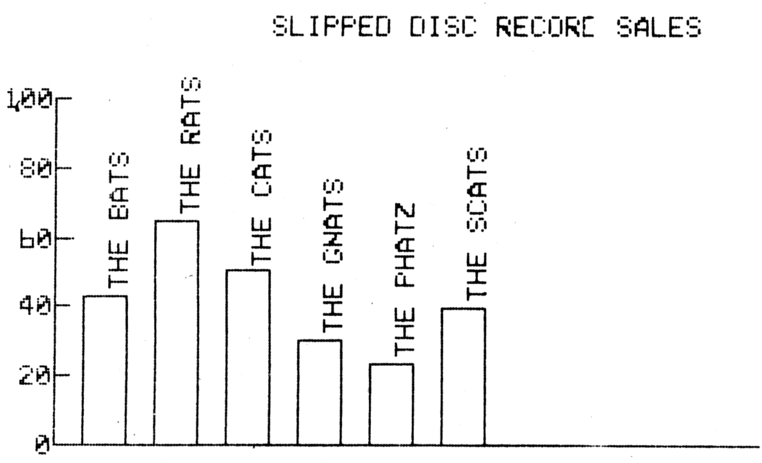


圖　5-5

可。主指示表消失了，然後兩軸出現在螢幕的繪圖區，在文字行部份，程式會問你資料項數。在你輸入之後，接著程式會問各個資料項的高度及標記，一旦你輸入某一資料項後，此長條圖便被畫出，當整個圖形完成後便一直保持在螢幕上直到你按下任一鍵。然後清除螢幕再重現主指示表以做下一圖形的輸入工作。

當然你也可藉改變選擇功能 1 到 4 來畫出不同的圖表。在你輸入一個選擇號碼後，螢幕出現其子指示表而你須依下列規則反應之：

1. TYPE 1 FOR VERTICAL, 2 FOR HORIZONTAL：打入數字 1 或 2 以說明長條圖的方向。
2. BAR WIDTH：輸入長條寬度（以螢幕點做單位）。
3. GAP BETWEEN BARS： 輸入各長條間隔（以螢幕點做單位）。
4. MAXIMUM VALUE REPRESENTED ：分別對垂直長條及水平長條輸入最大高度（以使用者單位計）。

   TICK MARK INTERVAL ：輸入間隔數（以螢幕點做單位）。

   MINIMUM CLEARANCE FOR CAPTIONS：輸入預留做文字標記的最小字元數。

BAR CHART 程式的選擇功能使你能畫出如圖 5-6A 及圖 5-6B 的長條圖。第一個圖用選擇功能 3 及 4 以將長條間隔減少為 0 並改變垂直軸的範圍。第二個圖用選擇功能 1 以將長條圖橫方向畫出。

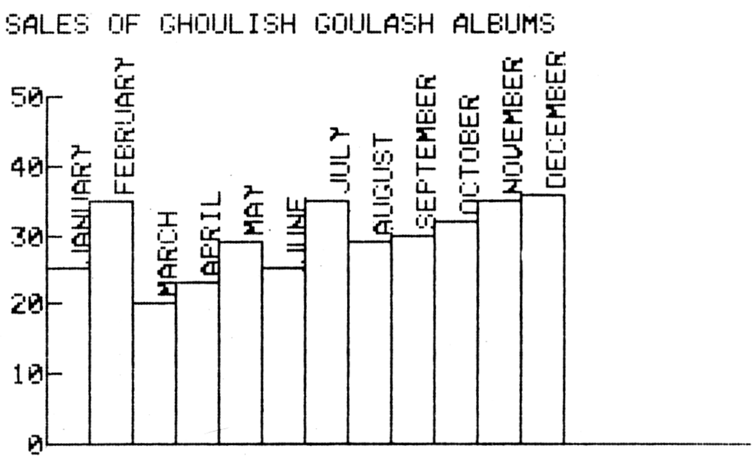


圖 5-6A

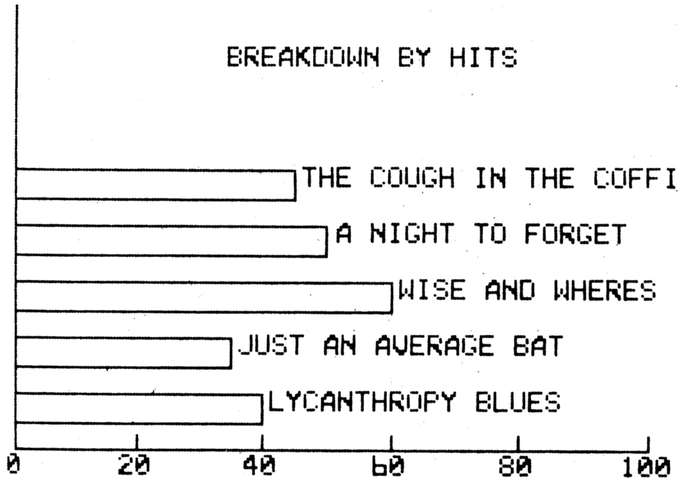


圖　5-6B

此程式使用的主要變數爲：

PT%(300，2)	用來定義字元檔的座標列陣，此在第 5-2 節解釋過。
DS%(200)	PT% 列陣的 Directory （參見 5-2 節）
DC%(70)	DS% 列陣的 Directory （參見 5-2 節）
BD	長條方向碼——1 代表垂直，2 代表水平
BW	長條寬度
BG	長條間隔寬
MV	在軸上標記的使用者最大值
TI	軸上格間隔
H	做爲標記及文字說明的字元高度

W	字元寬度
MC%	在水平長條右方或垂直長條上方預留文字空格的最小值
BD$(2)	使用者說明長條方向的VERTICAL或HORIZONTAL字串
OP	主指示表選擇號碼
XO,YO	圖表原點的螢幕座標
DE	延遲迴路索引變數
TT$	圖表名稱
XT,YT	圖表名稱開頭的螢幕座標
A$	等待使用者按下一鍵的假變數('dummy variable)
BC	相鄰垂直長條左邊或相鄰水平長條下邊的距離
SC	將使用者單位轉換爲螢幕座標的因素值——此值將使用者0到MV的範圍縮小(或放大)映射到螢幕範圍並能同時使螢幕顯示說明文字標記。
V	軸上某格的位置(使用者單位)
X,Y	軸上某軸的螢幕座標
X1	垂直長條的左邊 x 值
Y1	水平長條的下邊 y 值
VA	代表長條高或長度的變數
CA$	某長條的標記
XL,YL	此標記在螢幕上的起始座標
ST$	由劃字串副程式所畫出的文字串
M1,M2	文字或字串所寫出的方向(參見5-2節的FONT FORMER程式)

LL	某標記的字元數
MC	在長條上能容下的最大字元數

此程式中負責讀文字檔資料並顯示出字串的部份與FONT FORMER程式極爲相似，且使用同樣的變數，這些都已在 5-2 節解釋過。詳細的解釋可由各變數功能及各相對 REM 敍述內容了解之。

建議練習

1. 容許使用者能選擇各種不同形式的長條圖，如圖 5-7 所示。

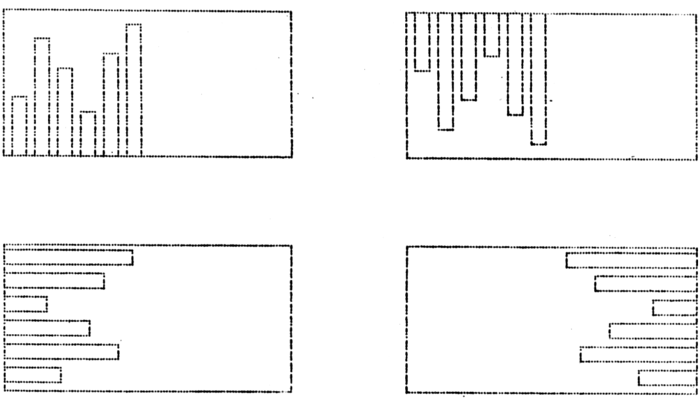


圖 5-7

2. 讓使用者說明以(A)填滿(B)空白(C)水平線(D)垂直線(E)點構成的虛線等方式將各長條內填滿。
3. 容許使用者選擇長條的那一端應做標記。
4. 使軸上的單位做標記。

5. 使整個圖形分格之。

6. 使整個圖表依軸的單位分格之。

7. 顯示出三維效果（圖 5-8 ）。

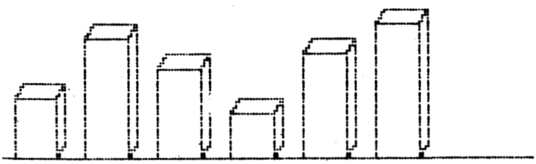


Figure 5.8

圖 5-8

8. 使長條深度代表另一變數（繼續第 7 項）。

9. 使間隔數為負值而造成重疊效果（圖 5-9 ）。

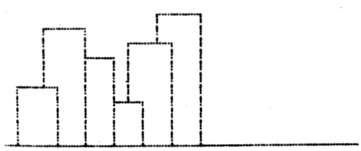


圖 5-9

10. 顯示出如圖5-10的圖表。

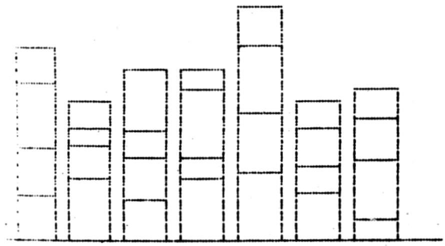


圖 5-10

11. 容許使用者定義畫長條圖的螢幕區。

12. 容許使用者能定義全部的圖形特性。

13. 使數個長條圖表帶不同的文字標記同時顯示在螢幕上。

BAR CHART 程式

```
100  REM *** BAR CHART ***
110  LOMEM: 16384: REM *** PROTECT MEMORY ***
120  DIM DC%(70),DS%(200),PT%(360,2): REM *** HI-RES FONT ***
130  GOSUB 3000: REM *** LOAD HI-RES FONT ***
140  BD=1:BW=15:BG=10:MV=100:TI=20: REM *** BAR & AXIS DATA ***
150  H=8:W=6:MC%=5
160  BD$(1)="VERTICAL":BD$(2)="HORIZONTAL"
170  X0=4*W:Y0=159-H
180  TEXT : HOME
190  PRINT  TAB( 10);"*** MAIN MENU ***"
200  PRINT  TAB( 05);"OPTION"; SPC( 20);"VALUE"
210  PRINT " "
220  PRINT "1. DIRECTION OF BARS"; TAB( 30);BD$(BD)
230  PRINT "2. WIDTH OF BARS"; TAB( 30);BW
240  PRINT "3. GAP BETWEEN BARS"; TAB( 30);BG
250  PRINT "4. AXIS DATA"; TAB( 30);MV;", ";TI;", ";MC%
260  PRINT "5. INPUT TITLE"
270  PRINT "6. INPUT DATA VALUES"
```

```
280  PRINT "7. QUIT"
290  VTAB 12: HTAB 5
300  CALL -958: REM *** CLEAR REST OF SCREEN ***
310  INPUT "PLEASE TYPE OPTION ";OP
320  ON OP GOTO 350,370,380,390,420,490,1010
330  PRINT "*** INVALID OPTION ***"
340  FOR DE=1 TO 1000: NEXT DE: GOTO 290
350  REM *** RESET PARAMETER VALUES ***
360  VTAB 13: INPUT "1 FOR VERTICAL, 2 FOR HORIZONTAL: ";BD: GOTO 18
370  INPUT "BAR WIDTH ";BW: GOTO 180
380  INPUT "GAP BETWEEN BARS ";BG: GOTO 180
390  INPUT "MAXIMUM VALUE REPRESENTED ";MV
400  INPUT "TICK MARK INTERVAL ";TI
410  INPUT "MINIMUM CLEARANCE FOR CAPTIONS: ";MC%: GOTO 180
420  VTAB 21: CALL -958
430  INPUT "TITLE FOR GRAPH: ";TT$
440  INPUT "X AND Y PLACEMENT: ";XT,YT
450  IF W*LEN(TT$)+XT<279 THEN 180
460  PRINT "TITLE TOO LONG-PRESS ANY KEY";
470  GET A$: GOTO 420
480  REM *** PLOT A BAR CHART ***
490  HGR : HCOLOR=3:BC=BG+BW
500  IF BD=2 THEN 630: REM *** BRANCH FOR HORIZONTAL BARS ***
510  REM *** BARS ARE VERTICAL ***
520  SC=(Y0-3-MC%*W)/MV: REM *** SETS SCALE FACTOR ***
530  REM *** DRAW VERTICAL AXIS AND BASE LINE ***
540  HPLOT X0,Y0-SC*MV TO X0,Y0 TO 279,Y0
550  REM *** DRAW TICK MARKS ON Y AXIS ***
560  FOR V=0 TO MV STEP TI: REM *** POSITION IN USER COORDINATES ***
570  Y=Y0-SC*V: REM *** POSITION IN SCREEN COORDINATES ***
580  HPLOT X0,Y TO X0+5,Y
590  GOSUB 1020
600  NEXT V: REM *** END OF Y TICK MARK CYCLE ***
610  GOTO 730: REM *** GET INPUT DATA ***
620  REM *** BARS ARE HORIZONTAL ***
630  SC=(279-X0-3-MC%*W)/MV: REM *** SETS SCALE FACTOR ***
640  REM *** DRAW VERTICAL STARTING LINE&HORIZONTAL AXIS ***
650  HPLOT X0,0 TO X0,Y0 TO X0+SC*MV,Y0
660  REM *** DRAW TICK MARKS ON X AXIS ***
670  FOR V=0 TO MV STEP TI: REM *** POSITION IN USER COORDINATES ***
```

```
680  X=X0+SC*V: REM *** POSITION IN SCREEN COORDINATES ***
690  HPLOT X,Y0 TO X,Y0-5
700  GOSUB 1020
710  NEXT V: REM *** END OF TICK MARK CYCLE ***
720  REM *** GET DATA TO PLOT ***
730  HOME : VTAB 22
740  INPUT "NUMBER OF BARS ";NB
750  IF NB=0 THEN 180
760  IF BD=2 THEN 870: REM *** BRANCH FOR HORIZONTAL BARS ***
770  REM *** BARS ARE VERTICAL ***
780  REM *** CYCLE ON X VALUE OF LEFT EDGE ***
790  FOR X1=X0+BG TO X0+BG+(NB-1)*BC STEP BC
800  VTAB 23: CALL -958
810  INPUT "VALUE, CAPTION: ";VA,CA$
820  REM *** PLOT A BAR ***
830  HPLOT X1,Y0 TO X1,Y0-SC*VA TO X1+BW,Y0-SC*VA TO X1+BW,Y0
840  GOSUB 1100: REM *** DRAW CAPTION ***
850  NEXT X1: REM *** END OF CYCLE FOR VERTICAL BARS ***
860  GOTO 970: REM *** GET THE OVERALL TITLE ***
870  REM *** BARS ARE HORIZONTAL ***
880  REM *** CYCLE ON Y VALUE OF LOWER EDGE ***
890  FOR Y1=Y0-BG TO Y0-BG-(NB-1)*BC STEP -BC
900  VTAB 23: CALL -958
910  INPUT "VALUE, CAPTION: ";VA,CA$
920  REM *** PLOT A BAR ***
930  HPLOT X0,Y1 TO X0+SC*VA,Y1 TO X0+SC*VA,Y1-BW TO X0,Y1-BW
940  GOSUB 1100: REM *** DRAW CAPTION  ***
950  NEXT Y1: REM *** END OF CYCLE FOR VERTICAL BARS ***
960  REM *** DRAW THE OVERALL TITLE ***
970  ST$=TT$:XL=XT:YL=YT
980  M1=1:M2=1: GOSUB 2500
990  HOME : VTAB 22: CALL -958
1000 PRINT "PRESS ANY KEY TO RETURN TO MAIN MENU": GET A$: GOTO 180
1010 END : REM *** TERMINATE ***
1020 REM *** DRAW TICK MARK LABEL ***
1030 ST$=LEFT$(STR$(V),4): REM *** CONVERT NUMBER TO A STRING ***
1040 LL=LEN(ST$)
1050 IF BD=1 THEN XL=X0-W*LL:YL=Y+3: REM *** ALIGN UNITS DIGIT ***
```

```
1060  IF BD=2 THEN XL=X-W*LL/2:YL=159: REM ***CENTER UNDER TICK MARK
1070  M1=1:M2=1
1080  GOSUB 2500: REM *** DISPLAY A STRING ***
1090  RETURN
1100  REM *** DISPLAY A CAPTION ***
1110  ST$=CA$
1120  IF BD=2 THEN 1210: REM *** BRANCH IF BARS ARE HORIZONTAL ***
1130  REM *** BARS ARE VERTICAL-FORM COORINATES OF FIRST LETTER ***
1140  XL=X1+INT(BW/2):YL=Y0-SC*VA-3
1150  MC=INT(YL/W): REM *** ALLOWED LENGTH ***
1160  IF MC<LEN(ST$) THEN ST$=LEFT$(ST$,MC)
1170  M1=4:M2=2: REM *** SIDEWAYS LETTERING MODE ***
1180  GOSUB 2500: REM *** DISPLAY THE STRING ***
1190  RETURN
1200  REM *** BARS ARE HORIZONTAL - FORM COORDS OF FIRST LETTER ***
1210  XL=X0+SC*VA+3:YL=Y1-INT(BW/2)
1220  MC=INT((279-XL)/W): REM *** ALLOWED LENGTH ***
1230  IF MC<LEN(ST$) THEN ST$=LEFT$(ST$,MC)
1240  M1=1:M2=1: REM *** HORIZONTAL LETTERING MODE ***
1250  GOSUB 2500: REM *** DISPLAY THE STRING ***
1260  RETURN
2500  REM *** DISPLAY A STRING ***
2510  IF LEN(ST$)<1 THEN  RETURN : REM *** NULL STRING ***
2520  FOR I=1 TO LEN(ST$): REM *** CYCLE FOR SUCCESSIVE CHARS  ***
2530  CN=ASC(MID$(ST$,I,1))-31: REM *** INDEX OF CHAR IN FONT DIRECTORY*
2540  S1=DC%(CN-1)+1:S2=DC%(CN): REM *** SEGMENT POINTERS ***
2550  P2=DS%(S1-1): REM ***DISPLAY A CHARACTER ***
2560  NS=S2-S1+1: GOSUB 3200
2570  NEXT I: REM *** END OF CYCLE FOR SUCCESSIVE CHARACTERS ***
2580  RETURN
3000  REM *** LOAD HI-RES CHARACTERS ***
3010  SW%=0
3020  W=6:H=8:A4$=CHR$(4):F1$="LABELTEXT"
3030  HOME : PRINT "FONT FILE(DEFAULT IS ";F1$;"): ": INPUT "";FL$
3040  IF LEN(FL$)=0 THEN FL$=F1$
3050  PRINT A4$;"OPEN ";FL$: PRINT A4$;"READ ";FL$
3060  INPUT "";NC: REM *** NUMBER OF CHARACTERS ***
3070  VTAB 12: FLASH : PRINT "READING IN DATA FROM ";FL$: NORMAL
3080  REM *** CYCLE THROUGH THE CHARACTERS ***
```

```
3090  S2=0:P2=0
3100  FOR CN=1 TO NC
3110  S1=S2+1
3120  INPUT "";NS: REM *** NUMBER OF SEGMENTS ***
3130  S2=S1+NS-1:DC%(CN)=S2
3140  GOSUB 3200: REM *** STORE ONE CHARACTER ***
3150  NEXT CN
3160  PRINT A4$;"CLOSE ";FL$: REM *** THE FONT HAS BEEN STORED ***
3170  SW%=1: REM *** SWITCH TO MAKE NEXT SUBROUTINE DISPLAY CHARS. ***
3180  RETURN
3190  REM *** STORE OR DISPLAY ONE CHARACTER ***
3200  IF NS<1 THEN 4700: REM *** NULL CHARACTER ***
3210  REM *** CYCLE THROUGH THE SEGMENTS ***
3220  FOR SN=S1 TO S2
3230  P1=P2+1: REM *** POINTS TO COORDINATE ARRAY ***
3240  IF SW%=1 THEN 3290: REM *** BRANCH TO DISLAY A CHARACTER
3250  REM *** STORE A CHARACTER ***
3260  INPUT "";NP: REM *** NUMBER OF POINTS ***
3270  P2=P1+NP-1:DS%(SN)=P2: REM *** STARTING POINTER ***
3280  GOTO 4210
3290  P2=DS%(SN):NP=P2-P1+1: REM ***HIGH POINTER TO DISPLAY
3300  REM *** CYCLE THROUGH THE POINTS ***
4210  FOR PN=P1 TO P2
4220  IF SW%<>0 THEN 4250: REM *** BRANCH TO DRAW A CHARACTER ***
4240  INPUT "";PT%(PN,1),PT%(PN,2): GOTO 4610: REM *** BRANCH TO
      STORE ONE ***
4250  REM *** DISPLAY A SEGMENT ***
4260  IF M2=2 THEN 4300: REM ***  BRANCH FOR SIDEWAYS CHARACTER ***
4270  IF NP=1 THEN  HPLOT XL+PT%(PN,1),YL+PT%(PN,2)
4280  IF PN>P1 THEN HPLOT XL+PT%(PN-1,1),YL+PT%(PN-1,2)TO XL+PT%(PN,1)
      YL+PT%(PN,2)
4290  GOTO 4610
4300  IF NP=1 THEN  HPLOT XL+H+PT%(PN,2),YL-PT%(PN,1)
4310  IF PN>P1 THEN  HPLOT XL+H+PT%(PN-1,2),YL-PT%(PN-1,1) TO XL+H+PT%
      (PN,2),YL-PT%
      (PN,1)
4610  NEXT PN,SN
4620  IF SW%=0 THEN  RETURN : REM *** CHAR. HAS BEEN STORED ***
4700  REM *** PROVIDE THE ESCAPEMENT FOR A DISPLACED CHARACTER ***
```

```
4710  IF M1=4 THEN 4810: REM *** BRANCH FOR SIDEWAYS WRITING ***
4720  XL=XL+W: RETURN : REM *** INCREASE X FOR NEXT CHARACTER ***
4810  YL=YL-W: RETURN : REM *** REDUCE Y FOR NEXT CHARACTER ***
5000  END
```

5—5 比率圖

前面說過在商用資料的圖形顯示法中比率圖佔很重要的地位，本節的 PIE CHART 便說明畫此類圖形的簡單原理。

首先顯示出主指示表，而各選擇值取爲內定值如下：

```
         *** MAIN MENU ***
    OPTION                  VALUE(S)
 1. POSITION OF CENTER  60,60
 2. RADIUS OF CHART     50
 3. TITLE OF CHART      SAMPLE PIE CHART
 4. POSITION OF TITLE   20,150
 5. POSITION OF KEY     120,20
 6. INPUT DATA
 7. QUIT

    PLEASE TYPE OPTION:
```

如果各內定值都適合你的要求，打入 6 。主指示消失後，螢幕上顯示出半徑爲50的圓，然後程式問你如果輸入單位爲百分比則打入 1 ，如爲實值則打入 2 。如果你打入 1 ，程式然後問分割整個圓成幾部份，接著依序要求輸入各部份的百分比及其文字標記，程式的畫法是依過圓心的水平半徑以反時針方向將各部份畫出，並在各部份分別標以號碼 1 ， 2 ，… ，等，同時在圖形右邊列出各部份的文字標記及其百分比，如圖5-11所示。

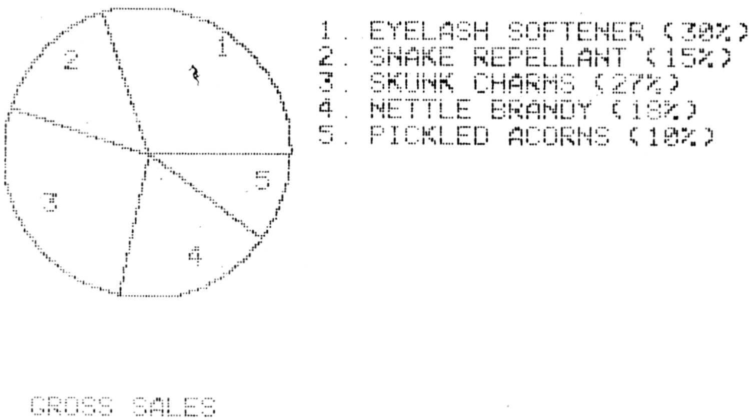


圖 5-11

如果百分比總數超過 100 ，則會顯示出一錯誤訊息。如果少於 100 ，剩下的部份將標記為 OTHER 。整個圖形一直保持在螢幕上直到你按下任何RETURN鍵，然後顯示出主指示表。

如果你輸入的是實際值而非其百分比，先打入 6 ，再打入 2 。程式便問你分割塊數，各部份的實際值，及其相對文字標記。當你輸入完這些資料後便將此圖畫出，同時也列出各部份的文字標記，如圖5-12所示

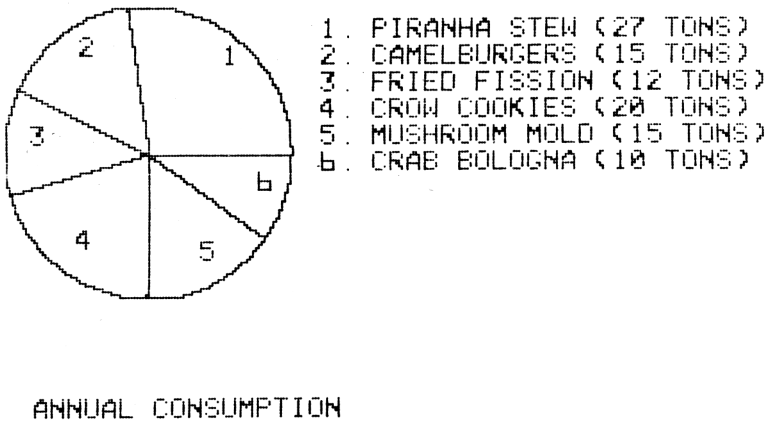


圖 5-12

此程式用到的主要變數爲：

PT%(300,2)	定義文字檔資料的座標，如第 5-2 節所示。
DS%(200)	PT% 列陣的 Directory（5-2 節）
DC%(70)	DS% 列陣的 Directory（5-2 節）
H,W	文字的高度及寬度
XC,YC	圓中心的螢幕座標
R	圓形半徑
TI$	圓形名稱
XT,YT	圓形名稱起始點之螢幕座標
XK,YK	key 開始處螢幕座標
F1,F2	常數 2π 及 π ／ 50

OP	主指示表選擇號數
A$	等待使用者按下的敍述裡的假變數
XL, YL	圖形裡某一半徑綫的終點座標
XO, YO	用來趨近圓弧的一短綫段起始座標
XN, YN	上述綫段的終點座標
TH	用來畫圓周或某塊圖的直綫終點的極角
MC	文字標記的最大字元數
XL, YL	文字串起點座標
ST$	文字串本身
TP	已輸入的累積百分比和
TP	現行畫圖段的極角
OZ	分別依使用者依百分比或實際值輸入而設定爲 1 或 2
NI	輸入的百分比個數或實際值個數
IN	處理各項資料的迴路索引
PE	上述資料項的百分比
CA$	上述資料項的識別名稱
MP	現行段的極角
UN$	當使用者以實際值輸入時之單位
SU	各連續項的累積實際值
AM(IN)	第 IN 項實際值

分析此程式的方法可利用上述的變數功能列表及程式裡的 REM 敍述。每一識別標記爲(1)代號，(2)一逗點，(3)文字標記（如果空間不夠便從中截斷）及(4)以括號表示百分比或實際量和其量度單位。

同樣的 PIE CHART 程式用 FONT FORMER 相同的方法來讀入並顯示字元及字串，這部份的變數使用法之解釋與第 5-2 節裡相同。同時BAR CHART 程式裡的 REM 敍述提供對文字檔更詳盡的解釋，這三個程式的部份程式裡的行數及敍述幾乎全部相同。

建議練習

1. 使程式當按 E 鍵時才完成圖形而不需在開始時輸入分割部份數目。
2. 容許使用者以第三種方法輸入資料，即先輸入總量，然後輸入各部份量，如果各部份量總和未及總量時將剩餘部份以 OTHER 表示。
3. 容許使用者控制第一條綫段與水平的夾角，如圖 5-13 裡即爲 −15°。

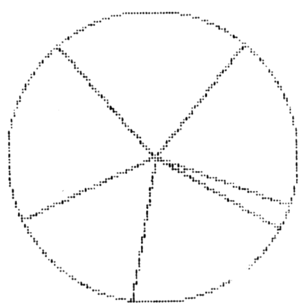


圖 5-13

4. 使各文字標記水平的放在(1)各段的裡面，(2)僅靠各圓弧外緣或(3)依 key 的方式輸出，此選擇由使用者控制。
5. 容許一個或數個部份移出一點點以強調其重要性（圖 5-14）。

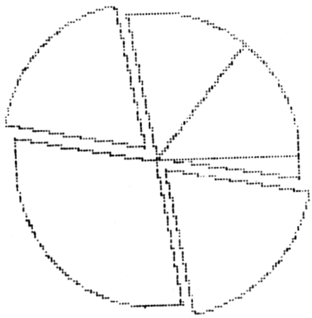


圖　5-14

6. 顯示出三維空間效果，如圖5-15所示。

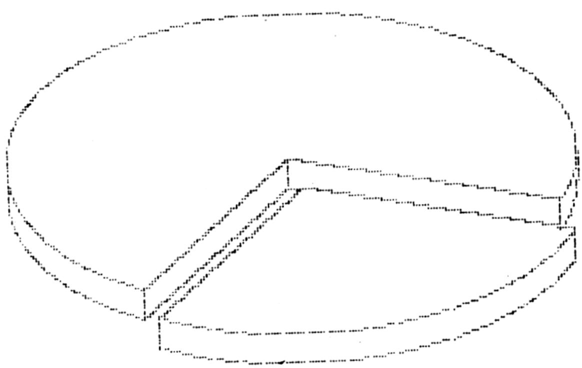


圖　5-15

7. 同時在螢幕上顯示出多個比率圖（圖 5-16 ）。

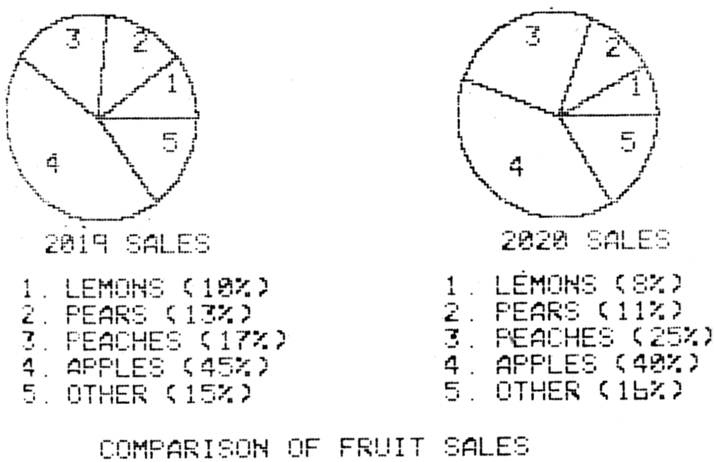


圖 5-16

PIE CHART 程式

```
100  REM *** PIE CHART ***
110  LOMEM: 16384: REM *** PROTECT MEMORY ***
120  DIM DC%(70),DS%(200),PT%(360,2):H=8:W=6: REM *** FOR HI-RES FON
130  GOSUB 3000: REM *** LOAD HI-RES FONT ***
140  XC=60:YC=60:R=50: REM *** DEFAULT CENTER AND RADIUS OF CHART **
150  TI$="SAMPLE PIE CHART":XT=20:YT=150: REM *** DEFAULT TITLE AND
     POSITION ***
160  XK=120:YK=20: REM *** DEFAULT POSITION OF KEY ***
170  F1=6.283:F2=.06283: REM *** 2 PI AND 2 PI/100 ***
180  TEXT : HOME
190  PRINT  TAB( 10);"*** MAIN MENU ***"
200  PRINT  TAB( 5);"OPTION"; SPC( 15);"VALUE(S)"
210  PRINT " "
220  PRINT "1. POSITION OF CENTER"; TAB( 23);XC;", ";YC
230  PRINT "2. RADIUS OF CHART"; TAB( 23);R
```

```
240  PRINT "3. TITLE OF CHART"; TAB( 23);TI$
250  PRINT "4. POSITION OF TITLE"; TAB( 23);XT;", ";YT
260  PRINT "5. POSITION OF KEY"; TAB( 23);XK;", ";YK
270  PRINT "6. INPUT DATA"
280  PRINT "7. QUIT"
290  VTAB 12: HTAB 5: CALL -958: REM *** CLEAR REST OF SCREEN ***
300  INPUT "PLEASE TYPE OPTION : ";OP
310  ON OP GOTO 340,350,360,370,380,390,1000
320  PRINT "INVALID OPTION :";OP
330  PRINT "PRESS ANY KEY TO CONTINUE": GET A$: GOTO 290
340  INPUT "COORDINATES OF CENTER: ";XC,YC: GOTO 180
350  INPUT "RADIUS OF CHART: ";R: GOTO 180
360  INPUT "TITLE: ";TI$: GOTO 180
370  INPUT "POSITION OF TITLE: ";XT,YT: GOTO 180
380  INPUT "POSITION OF KEY: ";XK,YK: GOTO 180
390  HGR : HCOLOR=3: REM *** PREPARE TO PLOT ***
400  IF XC>R AND YC>R THEN 420: REM *** BRANCH IF CHART WILL FIT ***
410  VTAB =21: PRINT "CHART WOULD NOT FIT ON THE SCREEN": GOTO 330
420  X0=XC+R:Y0=YC: REM *** STARTING POINT ON THE CIRCLE ***
430  FOR TH=0 TO F1 STEP F2: REM *** CYCLE AROUND THE CIRCLE ***
440  XN=XC+R*COS(TH):YN=YC-R*SIN(TH)
450  HPLOT X0,Y0 TO XN,YN: REM *** DRAW THE APPROXIMATION TO THE ARC
460  X0=XN:Y0=YN
470  NEXT TH
480  MC=INT((280-XK)/W-8): REM *** LIMIT THE LENGTH OF ENTRIES IN THE
     KEY ***
490  HPLOT XC,YC TO XC+R,YC: REM *** DRAW THE FIRST RADIAL LINE ***
500  PT=0: REM *** INITIALIZE THE ANGLE OF THE STARTING LINE OF THE
     SLICE ***
510  XL=XT:YL=YT:ST$=TI$: GOSUB 2500: REM *** DRAW THE TITLE ***
520  TP=0: REM *** INITIALIZE THE CUMULATIVE PERCENTAGE ***
530  VTAB 21: CALL -958: PRINT "TYPE 1 TO INPUT PERCENTAGES,"
540  INPUT "      2 TO INPUT AMOUNTS: ";OZ
550  IF OZ<>1 THEN 780: REM *** BRANCH TO TYPE AMOUNTS ***
560  VTAB 21: CALL -958: INPUT "NUMBER OF ITEMS: ";NI
570  FOR IN=1 TO NI: REM *** CYCLE FOR SUCCESSIVE SLICES ***
580  VTAB 22: CALL -958: INPUT "PERCENTAGE AND CAPTION: ";PT,CA$
590  TP=TP+PT: REM *** INCREASE THE CUMULATIVE PERCENTAGE ***
600  IF TP<=100 THEN 640: REM *** BRANCH IF VALID ***
610  VTAB 23: CALL -958: PRINT "TOTAL EXCEEDS 100 PERCENT"
620  PRINT "PRESS ANY KEY TO RETYPE LAST ENTRY"
630  TP=TP-PT: GET A$: GOTO 580: REM *** TRY AGAIN ***
```

```
640  TH=TP*F2: REM *** ANGLE IN RADIANS OF OUTER EDGE OF CURRENT SLIC
650  HPLOT XC,YC TO XC+R*COS(TH),YC-R*SIN(TH): REM *** DRAW IT ***
660  MP=TH-PT*F2/2: REM *** ANGLE OF MEDIAN LINE OF SLICE ***
670  XL=XC+(R-9)*COS(MP):YL=YC-(R-9)*SIN(MP): REM *** COORDS OF REF N
680  ST$=STR$(IN): GOSUB 2500: REM *** DRAW THE REFERENCE NUMBER ***
690  XL=XK:YL=YK+(IN-1)*(H+1): REM *** COORDINATES OF KEY ***
700  ST$=ST$+". "+LEFT$(CA$,MC)+" ("+STR$(PT)+"%)": REM *** FORM THE
710  GOSUB 2500: REM *** DRAW IT ***
720  PT=TP: REM *** RESET ANGLE OF STARTING LINE FOR NEXT SEGMENT ***
730  NEXT IN:: REM *** END OF CYCLE FOR SUCCESSIVE SLICES ***
740  VTAB 22: CALL -958: REM *** MOVE CURSOR AND CLEAR THE SCREEN ***
750  PRINT "PRESS ANY KEY TO RETURN TO MAIN MENU": GET A$
760  GOTO 180
770  REM *** THE USER INPUTS THE AMOUNTS ***
780  INPUT "UNITS: ";UN$:MC=MC-LEN(UN$)-2: REM *** ALLOW FOR UNITS IN
790  INPUT "NUMBER OF ITEMS: ";NI
800  SU=0:PT=0:REM *** INITIALIZE THE TOTAL ***
810  FOR IN=1 TO NI: REM *** CYCLE FOR SUCCESSIVE AMOUNTS ***
820  VTAB 23: CALL -958: INPUT "AMOUNT AND CAPTION: ";AM(IN),CA$(IN)
830  SU=SU+AM(IN): REM *** INCREASE THE TOTAL&FORM THE KEY ***
840  CA$(IN)=STR$(IN)+". "+LEFT$(CA$(IN),MC)+" ("+STR$(AM(IN))+" "+UN
850  NEXT IN: REM *** END CYCLE FOR SUCCESSIVE AMOUNTS ***
860  FOR IN=1 TO NI: REM *** CYCLE FOR SUCCESSIVE SLICES ***
870  TP=TP+(AM(IN)/SU*100): REM *** INCREASE THE CUMULATIVE PERCENTAG
880  TH=TP*F2: REM *** CONVERT TO RADIANS ***
890  HPLOT XC,YC TO XC+R*COS(TH),YC-R*SIN(TH): REM *** DRAW THE EDGE
900  MP=(TH+PT)/2: REM *** ANGLE OF MEDIAN LINE OF SLICE ***
910  XL=XC+(R-9)*COS(MP): REM *** COORDINATES FOR ENTRY IN KEY ***
920  YL=YC-(R-9)*SIN(MP)
930  PT=TH: REM *** FIRST EDGE FOR NEXT SLICE ***
940  ST$=STR$(IN): GOSUB 2500: REM *** DRAW THE SLICE NUMBER ***
950  XL=XK:YL=YK+(IN-1)*(H+1): REM *** COORDS OF ENTRY IN KEY ***
960  ST$=CA$(IN): GOSUB 2500: REM *** DRAW THE ENTRY ***
970  NEXT IN: REM *** END OF CYCLE FOR SUCCESSIVE SLICES ***
980  PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: GOTO 180
1000 END : REM *** QUIT ***
2500 REM *** DISPLAY A STRING ***
2510 IF LEN(ST$)<1 THEN  RETURN
2520 FOR I=1 TO LEN(ST$)
2530 CN=ASC(MID$(ST$,I,1))-31
```

```
2540  S1=DC%(CN-1)+1:S2=DC%(CN)
2550  P2=DS%(S1-1)
2560  NS=S2-S1+1: GOSUB 3200
2570  NEXT I
2580  RETURN
3000  REM *** LOAD AND DISPLAY CHARACTERS ***
3010  SW%=0
3020  W=6:H=8:A4$=CHR$(4):F1$="LABEL.EXT"
3030  HOME : PRINT "FONT FILE (DEFAULT IS ";F1$;"): ";: INPUT "";FL$
3040  IF LEN(FL$)=0 THEN FL$=F1$
3050  PRINT A4$;"OPEN ";FL$: PRINT A4$;"READ ";FL$
3060  INPUT "";NC
3070  VTAB 12: FLASH : PRINT "READING DATA FROM ";FL$: NORMAL
3080  REM *** CYCLE THROUGH THE CHARACTERS ***
3090  S2=0:P2=0
3100  FOR CN=1 TO NC
3110  S1=S2+1
3120  INPUT "";NS
3130  S2=S1+NS-1:DC%(CN)=S2
3140  GOSUB 3200
3150  NEXT CN
3160  PRINT A4$;"CLOSE ";FL$
3170  SW%=1
3180  RETURN
3190  REM *** STORE OR DISPLAY ONE CHARACTER ***
3200  IF NS<1 THEN 4700
3210  REM : CYCLE THROUGH THE SEGMENTS
3220  FOR SN=S1 TO S2
3230  P1=P2+1
3240  IF SW%=1 THEN 3290
3250  REM *** STORE A CHARACTER ***
3260  INPUT "";NP
3270  P2=P1+NP-1:DS%(SN)=P2
3280  GOTO 4210
3290  P2=DS%(SN):NP=P2-P1+1
3300  REM *** CYCLE THROUGH THE POINTS ***
4210  FOR PN=P1 TO P2
4220  IF SW%<>0 THEN 4250
```

```
4240  INPUT "";PT%(PN,1),PT%(PN,2): GOTO 4610
4250  REM *** DISPLAY A SEGMENT ***
4270  IF NP=1 THEN  HPLOT XL+PT%(PN,1),YL+PT%(PN,2)
4280  IF PN>P1 THEN HPLOT XL+PT%(PN-1,1),YL+PT%(PN-1,2)TO XL+PT%
     (PN,1),YL+PT%(PN,2)
4610  NEXT PN,SN
4620  IF SW%=0 THEN  RETURN
4700  REM *** PROVIDE THE ESCAPEMENT ***
4720  XL=XL+W: RETURN
5000  REM ***  END  ***
```

第六章　曲線圖形

此章我們將由僅用直綫和圓的圖形應用推展到其他種類的曲綫圖形。拋體運動的軌跡就是其中一個很好的例子，實際上此曲綫即為數學上的拋物綫軌跡，而同時也顯示了數學在日常生活中的應用之廣泛。本章裡的其他曲綫在工程式做機械或其他工程設計中也會用到，例如人造或自然衛星的軌跡實際上就是數學裡的橢圓方程式，其他如擺綫也常用在彈性學理論中，正弦函數曲綫的應用在電學及許多物理現象裡也經常出現。所有這種曲綫本章都有列出繪圖程式。

6−1　拋物體軌跡

如果你想應用拋體軌跡或寫一個類似此曲綫的遊戲，本節的GALIEO程式可能有些幫助。此程式畫出由地面上以一特定仰角拋出的物體運動軌跡。程式要求輸入⑴仰角⑵拋體大小（亦即代表此拋體的正方形螢幕點數）及⑶初速度。此程式畫出的圖形在初速為每秒 110 呎，仰角 66.5° 時，正好將曲綫頂度觸及螢幕頂部而其落地點正好在螢幕右邊。表現運動軌跡

的方法是依拋物線將一小方塊在螢幕上畫出然後再消失以造成連續動作，圖6-1A到6-1C便是典型的圖形。

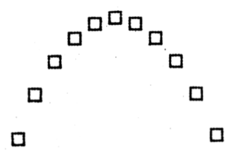
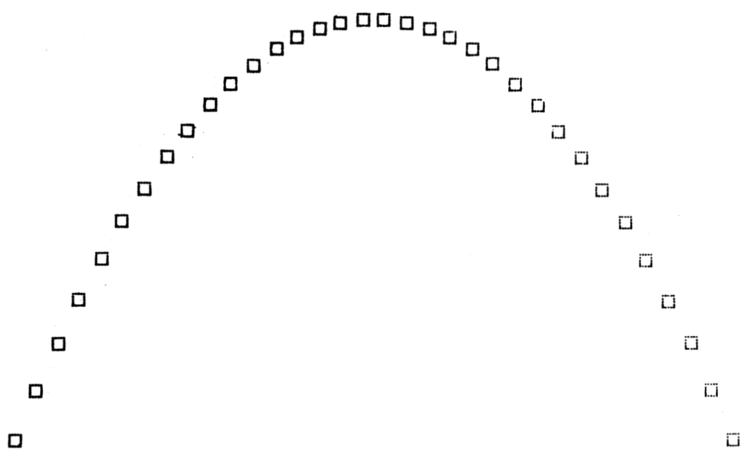


圖 6-1A Default.



v=105.

圖 6-1B

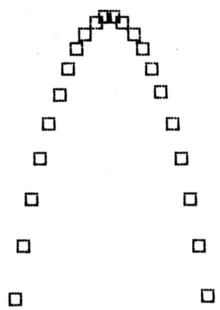


圖　6-1C　v=80, angle=80°.

在畫完一個拋體軌跡後，程式會將螢幕清除然後再重新要求輸入新圖形資料，你可以不同的仰角及初速度試驗各種圖形的結果。此程式並未考慮其他如風阻力的因素，所以距離範圍及最大高度都完全取決於 v 。是故如果你令 $v = 55.5$ 呎／每秒的話，同時保持仰角爲 66.5° 時，軌跡的最大高度僅及螢幕的一半高度，同時在螢幕的一半處便結束其運動軌跡。

改變仰角的效果取決於簡單的三角理論。如果 $v = 100$ ，而對各種不同仰角的相對範圍及高度列於下：

角度：	0	10	20	30	40	50	60	70	80	90
範圍：	0	106	200	269	306	306	269	200	106	0
高度：	0	5	18	39	64	91	116	137	150	155

如果你想一探究竟，此公式是非常簡單的。如果 a 代表仰角，g 代表重力常數，t 代表時間，則

範圍　$R = v^2 \sin 2a / g$

最大高度　$H = (v \sin a)^2 / 2g$

在 t 秒時 x 值：　$v t \cos a$

在 t 秒時 y 值：　$v t \sin a - g t^2 / 2$

y 座標以 x 座標表示則爲：　$y = (4H / R^2) x (x - R)$

此程式所用到的主要變數爲：

A	仰角
V	初速度
S	抛體大小
G	重力常數 g
HG	g 值一半
CF	由度度量轉換爲弳度量的因素
AR	仰角以弳度量爲單位之值
OP	主指示表選擇號數
HC，VC	初速度的水平及垂直分量
TF	飛行時間
HS	抛體一邊之半（以螢幕單位表示）
DT	模擬飛行時間
T	某一飛行時間
CO	顏色——畫抛體時爲 3，擦去時爲 0
K	控制模擬速度的迴路索引

在顯示完各要求資料訊息後，程式便經歷一段以等時間間隔來計算抛

體連續位置的週期，並相對的在螢幕上顯示出來，同時將舊位置上的拋體擦去，其中 x ，y 值都直接由上述各公式計算得之。

建議練習

1. 在螢幕左邊依指定方向畫出一大砲並發射出砲彈（此可利用來做遊戲設計）。
2. 使拋體由螢幕右邊飛到左邊。
3. 對螢幕上一固定不動目標發射一拋體，如果擊中的話則發出聲響且此目標消失（參見 2-4 節的 ROBOT 程式）。
4. 將上述目標以小塊碎裂而消失。
5. 將上述目標在被擊中時在螢幕上到處亂動。
6. 使用者能控制第 1 項的砲口。
7. 畫兩大砲而在螢幕上相對開火。
8. 將拋體畫成一球，而拋向螢幕右邊的網中。
9. 使用者能移動第 8 項的網子。
10. 使球在落地時以原仰角及初速反彈。
11. 使用者輸入一比率，而球在每反彈一次時依此比率減低其速度。
12. 利用圖形表來定義此拋體形狀。
13. 在螢幕上由上到下投射出一連串拋體。
14. 如果你有搖捍的話，用它來控制拋體的仰角及初速。

GALILEO程式

```
100  REM *** GALILEO-PLOTS A TRAJECTORY ***
110  A=66.5:V=55.5:S=4: REM *** DEFAULT LAUNCH ANGLE, VELOCITY AND SIZE ***
120  G=32.2:HG=G/2:CF=57.2958: REM *** PHYSICAL CONSTANTS ***
130  POKE 34,23: REM *** PROTECT THE MENU FROM SCROLLING ***
140  TEXT : HOME : HGR: HCOLOR=3
150  REM *** MAIN MENU ***
160  VTAB 21: CALL -958: REM *** CLEAR THE TEXT AREA ***
170  PRINT "*** OPTIONS AND CURRENT VALUES ***"
180  PRINT "1. ANGLE (";A;"), 2. VELOCITY (";V;")"
190  PRINT "3. SIZE (";S;"), 4. DISPLAY PATH, 5. QUIT"
200  REM *** PROMPT FOR INPUT ***
210  VTAB 24: CALL -958: INPUT "TYPE AN OPTION? ";OP
220  VTAB 24: CALL -958: REM *** ERASE THE LAST LINE ***
230  ON OP GOTO 260,270,280,290,430
240  VTAB 24: PRINT "INVALID OPTION"
250  FOR X=1 TO 1000: GOTO 210
260  INPUT "FIRING ANGLE? ";A: GOTO 160
270  INPUT "VELOCITY? ";V: GOTO 160
280  INPUT "SIZE OF PROJECTILE? ";S: GOTO 160
290  REM *** COMPUTE DATA FOR PROJECTILE'S PATH ***
300  AR=A/CF:HC=V*COS(AR):VC=V*SIN(AR)
310  IF S>8 THEN S=8
320  TF=VC/HG:HS=S/2
330  DT=TF/(2+INT(V*TF/20))-.0001: REM *** ALLOW 20 SITINGS ***
350  FOR T=0 TO TF STEP DT: REM *** SIMULATE THE PASSAGE OF TIME ***
360  X=5+T*HC
370  Y=154-T*(VC-T*HG)
380  CO=3: GOSUB 440: REM ***  DRAW THE PROJECTILE ***
390  FOR K=1 TO 50: NEXT K: REM *** LEAVE IT ON THE SCREEN BRIEFLY ***
400  CO=0: GOSUB 440: REM *** ERASE IT ***
410  NEXT T: REM *** END THE SIMULATION CYCLE ***
420  GOTO 150
430  TEXT : HOME : END
440  HCOLOR=CO: REM *** DRAW OR ERASE THE PROJECTILE ***
450  IF X-HS<0 OR X+HS>279 THEN  POP : GOTO 160: REM *** START AGAIN ***
460  IF Y-HS<0 OR Y-HS>159 THEN  RETURN : REM *** BYPASS THIS VALUE ***
470  HPLOT X-HS,Y-HS TO X-HS,Y+HS TO X+HS,Y+HS TO X+HS,Y-HS TO X-HS,Y-HS
480  FOR K=1 TO 100: NEXT K: REM *** DELAY BEFORE THE NEXT ACTION ***
490  RETURN
```

6—2 上下移動的曲線

本節的 POLYPLOT 程式不但可畫出前一節的拋物綫軌跡，同時也可畫出其種種上下移動的多項式圖形，圖 6-2A 到圖 6-2D 便是其中的一些例子。其中圖形的 *y* 值可能由 *x* 值的一次，二次，三次或多次方計算出來，甚至爲將各項相加的總和，這種方程式我們稱之爲多項式。

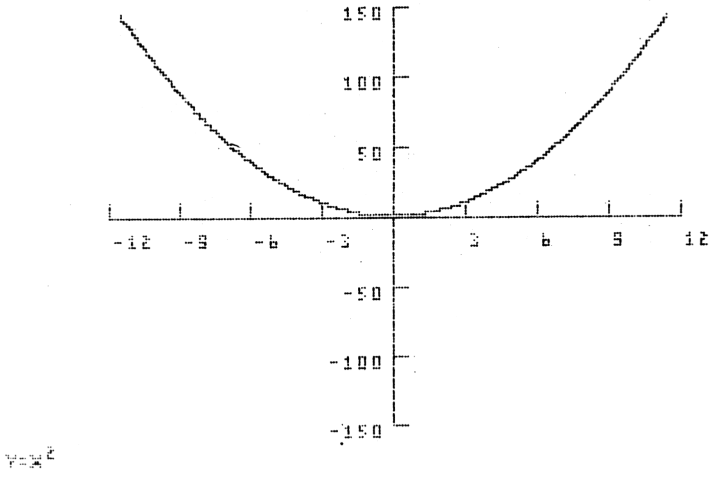


圖 6-2A

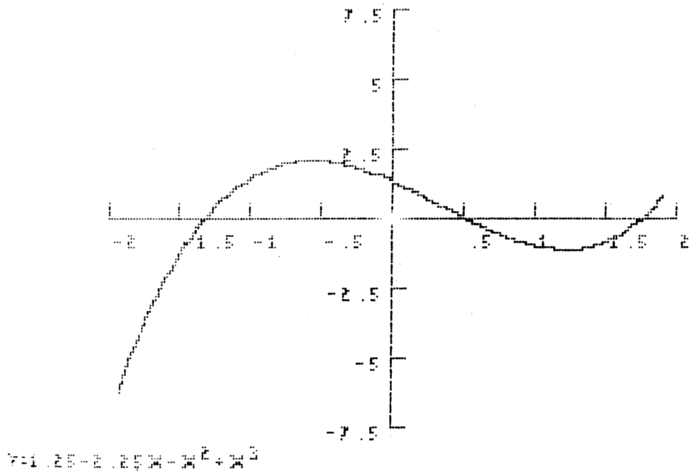


圖 6-2B

此程式首先顯示出主指示表；

```
TYPE 1 FOR SAMPLE POLYNOMIALS
     2 TO SPECIFY A POLYNOMIAL
     3 TO QUIT:
```

如果你打入 1 ，程式會繼續提供另一指示表：

```
TYPE 1 FOR QUADRATICS,  2 FOR CUBICS,
     3 FOR OTHERS,  4 FOR MAIN OPTIONS:
```

如果你打入 1 ，則程式顯出方程式 $y = 3 + 2x + 0.1x^2$ 的圖形（圖 6-3），其中 x 值範圍由 -48 到 $+48$ ，y 值範圍由 -60 到 $+60$ ，同時 x 軸分為 8 等份而以其相對數值標記之，y 軸也是如此。圖形的左下方則列出

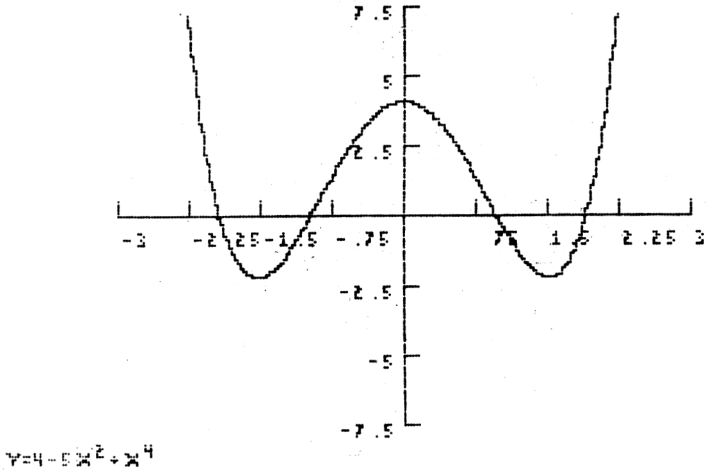

7.5
5
2.5
-2.5
-5
-7.5
-3
-2.25
-1.5
-.75
.75
1.5
2.25
3
Y=4-5X^2+X^4

圖　6-2C

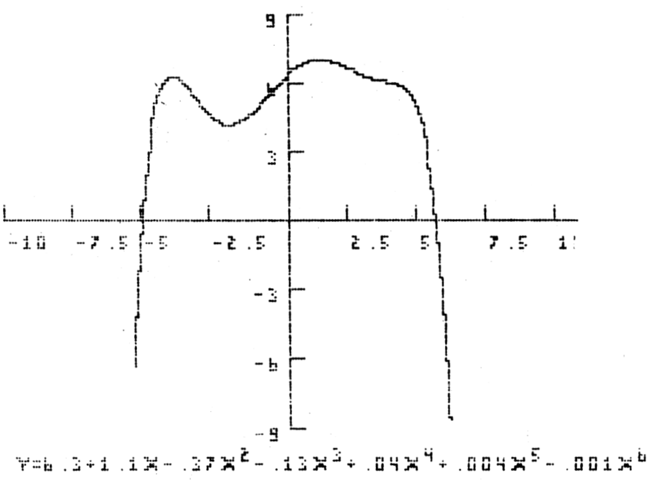

9
3
-3
-6
-9
-10
-7.5
-5
-2.5
2.5
5
7.5
Y=6.3+1.1X-.37X^2-.13X^3+.04X^4+.004X^5-.001X^6

圖　6-2D

此方程式。

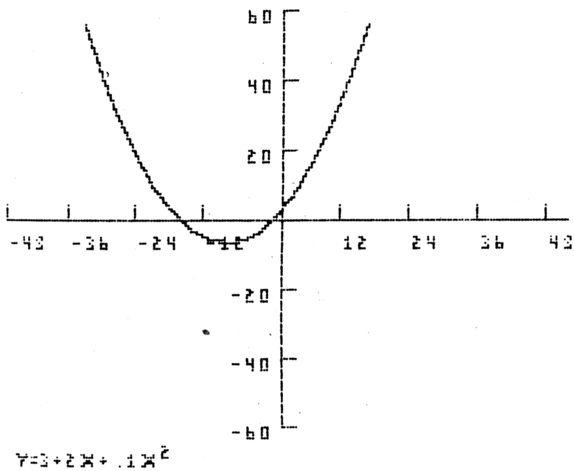


圖 6-3

然後程式顯示出：

```
TYPE 1 FOR ANOTHER QUADRATIC,
     2 TO RETURN TO MAIN MENU:
```

如果你打 1 ，則程式畫出方程式 $y = 1 - 2x - 2x^2$ 的圖形（圖6-4 ）。此時 x 軸範圍由 -8 到 $+8$ ， y 軸範圍由 -30 到 $+30$ ，同樣的 x 及 y 軸都等分為 8 個間隔並以相對的數值標記之。

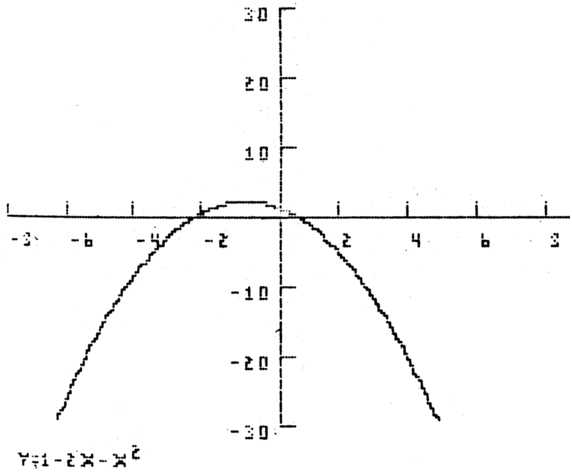


圖 6-4

如果程式繼續顯示指示表時你再繼續打入 1 ，則畫出的圖形順序爲圖 6-5A 到圖 6-5D ，其中都有適當的數值標記及所用的方程式。如果你繼續輸入 1 以顯示二次方程式圖形，則程式所畫出的圖形便回到第一個圖，即圖 6-3（因爲程式本身只記憶這些範例方程式）。

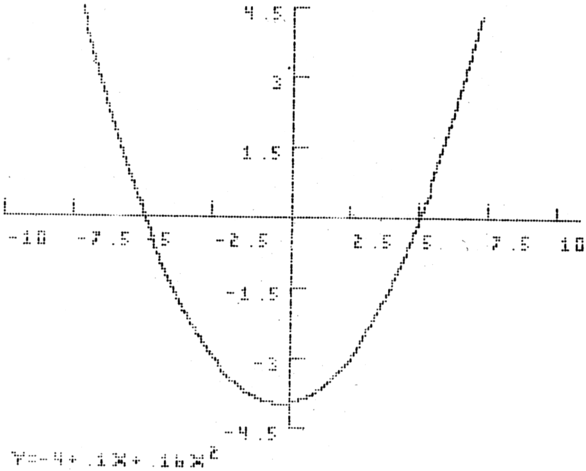

4.5
3
1.5
-10
-7.5
-5
-2.5
2.5
5
7.5
10
-1.5
-3
-4.5
Y=-4+.1X+.16X²

圖 6-5A

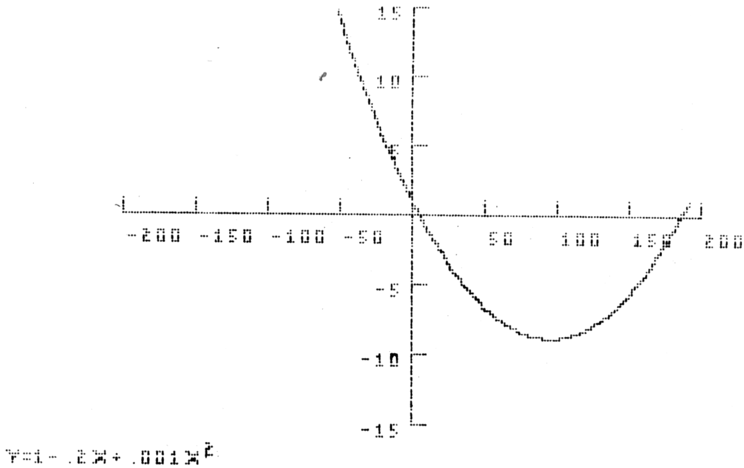

15
10
5
-200
-150
-100
-50
50
100
150
200
-5
-10
-15
Y=1-.2X+.001X²

圖 6-5B

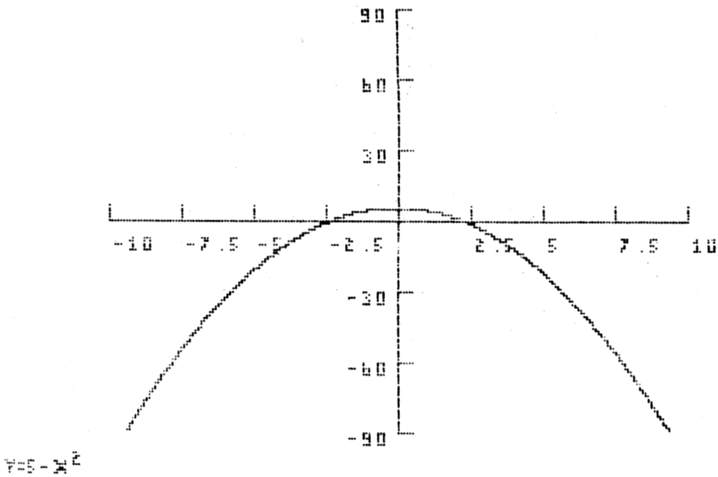

90
60
30
-10
-7.5
-5
-2.5
2.5
5
7.5
10
-30
-60
-90
Y=5-X²

圖　6-5C

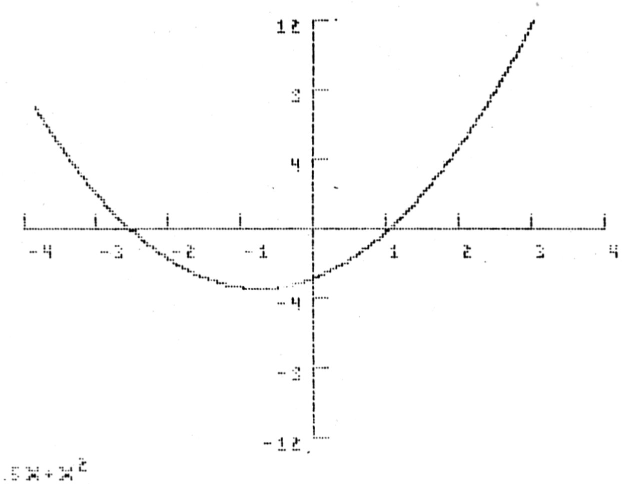

12
8
4
-4
-3
-2
-1
1
2
3
4
-4
-8
-12
.5X+X²

圖　6-5D

如果你(1)打入 2 回到主指示表，(2)在主指示表選擇 1 以繼續看一些多項式圖形，(3)在次指示表裡選擇代表三次方程式的 3 ，則程式會顯示出圖 6-6 的三次方程式圖形，同時列出下列訊息：

```
TYPE 1 FOR ANOTHER CUBIC,
    2 TO RETURN TO MAIN MENU:
```

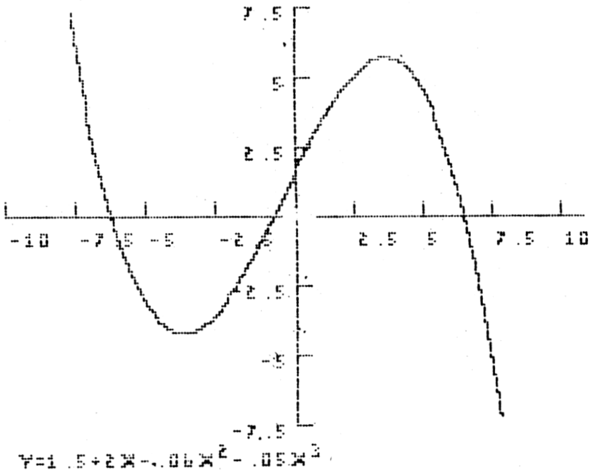


圖 6-6

每當此訊息出現時，如果你都選擇 1 ，便可得到如圖 6-7A 到圖 6-7C 的三次方程式圖形。這些曲綫有的含兩次平滑區（如圖 6-7A 和圖 6-7B ），有的僅含一個平滑區（如圖 6-7C ）。這些圖形皆因 y 與 x 的關係最高爲三次方，故稱之爲三次曲綫（ cubics ）。

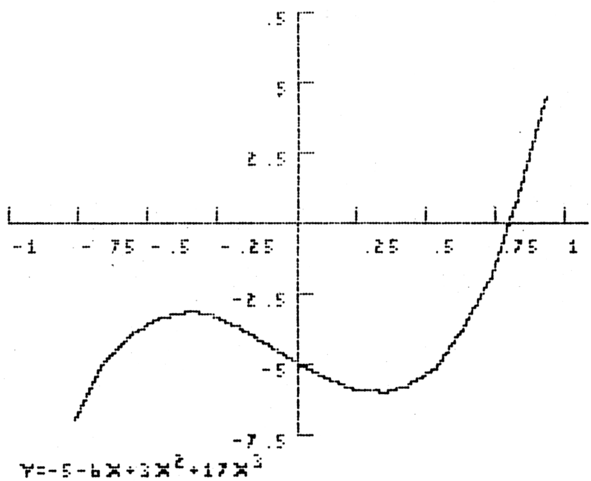


圖 6-7A

如果你(1)輸入 2 以回到主指示表，(2)在次指示表裡繼續選擇 1 以多看一些多項式範例圖，(3)在次指示表裡選擇 3 ： OTHERS，則程式畫出如圖 6-8 的圖形，同時顯示出下列訊息：

```
TYPE 1 FOR ANOTHER POLYNOMIAL,
     2 TO RETURN TO MAIN MENU:
```

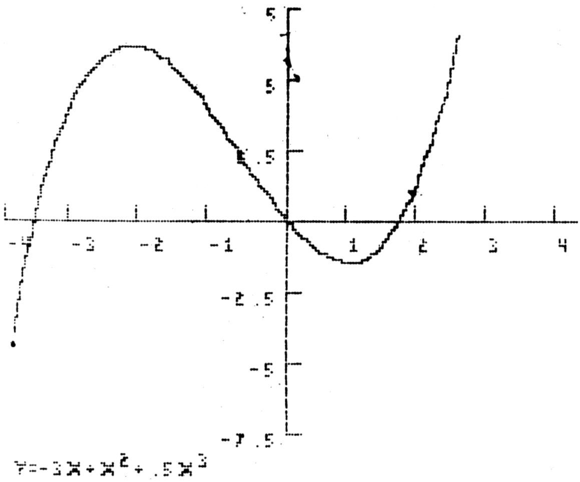


圖 6-7B

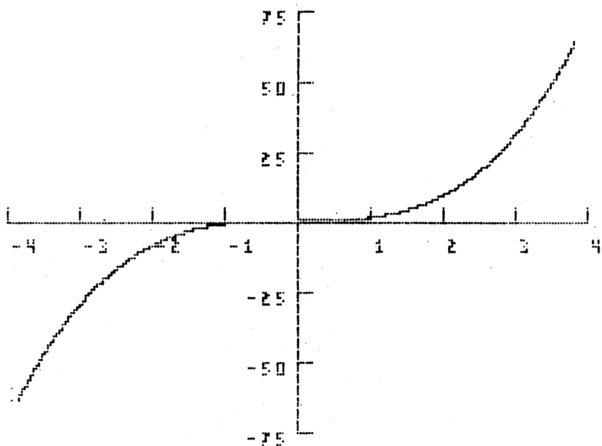


圖 6-7C

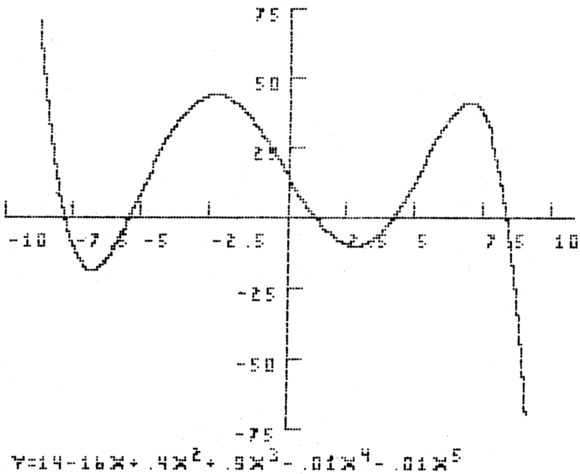


圖 6-8

如果你繼續選擇範例圖形，則可得到一連串其他上下移動的多項式曲線，直到你滿意爲止。

如果你想定義一些自己的多項式，輸入 2 以回到主指示表。程式便會要求你輸入此多項式的次數（即最高次方數），各項的係數常數項，x 和 y 軸的間隔值，及畫圖時 x 使用的間隔。然後程式便依輸入的 x 間隔乘以 4 後取其正負值做爲 x 的範圍（正如圖 6-2A 中的間隔爲 3，而 x 取值範圍爲 -12 到 $+12$），y 值的範圍則爲所取 y 間隔值的三倍正負值，如果圖形超過 y 值的範圍便會自動切掉。

在畫出自行定義的多項式後，程式繼續顯示：

```
TYPE 1 TO SPECIFY ANOTHER POLYNOMIAL,
     2 TO RETURN TO MAIN MENU:
```

如果你打入 1 ，則又回到前面討論圖形裡。如果你打 2 ，則回到主指示表。

此程式所用到的主要變數分別爲：

CO(IQ)	第 IQ 個二次方程式的常數項
C1(IQ)	第 IQ 個二次方程式的 x 係數
C2(IQ)	第 IQ 個二次方程式的 x^2 係數
AX(IQ)	畫圖時 x 軸的間隔值，亦即 x 最大值的 1 ／ 4
BY(IQ)	畫圖時 y 軸的間隔值，亦即 y 最大值的 1 ／ 3
D(IQ)	畫圖時 x 所取的描點間隔
H0(IQ), H1(IQ), H2(IQ), H3(IQ)	第 IQ 個三次方程式的常數項及各係數
EX(IQ), FY(IQ), G(IQ)	畫此三次方程式 x 軸，y 軸間隔及 x 所取描點間隔
JX(IQ), KY(IQ), GG(IQ)	相對於第 IQ 個更高次多項式的各項資料
CD(IR)	正在畫圖多項式的第 IR 個係數
IQ	程式用來示範同類方程式的索引號碼
CC(IR)	表示更高方程式的第 IR 個數——高次方程式的表示法是以其次數做開頭，後隨常數，一次項係數，直到最高次係數，其中缺項者應填 0 。
IM(1), IM(2), IM(3)	分別代表示範的二次，三次及高次方程式數目
IR	CC 列陣索引
DG%	多項式的次數
IS	各連續項的索引值，以常數項爲 0 做開始

WD$(1),WD$(2),WD$(3)	
	分別代表 QUADRATIC, CUBIC 及 POLYNOMIAL
W, H	畫出的字元寬及高度
HH	上述高度之半
OP	使用者選擇主指示表的功能數或選擇顯示另一多項式的功能數
OQ	使用者在次指示表裡用來說明顯示多項式的次數
XD, YD	現行畫圖中 *x* 及 *y* 軸的間隔
DP	現行畫圖中 *x* 的描點間隔
TN	使用者定義自己的多項式週期裡的係數索引
XL, YL	下一字元在螢幕上的座標── 畫上後 XL 增加一字元寬度
SW%	一控制是否要顯示多項式前的 + 號的變數
XB%	現行項起點在螢幕上的 *x* 位置── 程式如果發現無足夠空間以完成此項時便追踪回此點
NT	在畫多項式於螢幕上的週期中以常數項爲起始點的係數索引
ST%	欲寫出的字元串
I	上述字元串的字元索引
EX, MA	浮點數中的指數及小數部份。
II	在將小於 1 的浮點數轉換爲一般小數型式時在有意義位數前放適當數目個的零的週期之索引
SX, SY	將使用者單位轉換爲螢幕座標之因素
XU, YU	欲畫之點的 *x*，*y* 座標(以使用者單位計)
XN, YN	上述點相對的螢幕座標

XP , YP	前一點的螢幕座標
OP%	在畫完一多項式後使用者用來選擇各種功能的代號
XS , YS	x , y 軸的螢幕 x , y 座標
NN	畫出標記的迴路索引
CH$	在顯示字串時正在畫出的字元

本節程式的列表裡，大部份的 REM 敍述都是寫好程式後再加上去的，其主要目的是幫助你了解此程式。所以在輸入此程式應將各個 REM 敍述去掉，因爲整個程式的體積太大而無法在Apple II 上執行。但需保留空的 REM 敍述，此有時做爲其他敍述轉移到的地方。你必須確實而正確的將 REM 敍述內容除去，以使程式能順利執行。

POLYPLOT 程式首先從DATA 敍述裡讀入預先設定好的多項式資料到適當的列陣中，接著顯示主指示表。視使用者之選擇，程式將適當的多項式資料放到變數裡或直接由使用者輸入資料，無論要畫的是程式自定或是由使用者定義的多項式，將適當變數設定好後，程式便將曲綫畫出，顯示字元的方法是使用一些含簡單 HPLOT 敍述的副程式將構成字元的綫段畫出來，如果要顯示的字元數較少時，這種方法較前面所述的FONT FORMER 程式方法簡單的多。

程式裡也做一些簡單的檢查以除去零項，項數前的加號，及避免係數或指數爲 1 的乘法。同時也需將浮點數轉換爲一般表示法，STR$ 函數將數目轉換成字串，使其能用 Apple II 有力的字串處理函數來處理資料。

此程式的畫圖區是在螢幕水平40到 240 點，垂直 0 到 150 點之間，兩軸交點在此區域正中央。各軸的間格是依距離原點 XD 或 YD 的整數倍標明之，而其文字標記則用前面提過的叫數個含 HPLOT 副程式來執行。SX 及 SY 是用來將使用者的單位轉換爲螢幕座標的兩項因素。

多項式的計算法是用標準的計算多項式方法論（ Algorithm ），此方法即以最高次係數開始，乘以 x 值後再加上次高項係數，再乘以 x 值，再加上次一項係數，如此重覆加到常數項爲止。所以一個 n 次多項式

$$y = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots\cdots + a_0$$

可表爲

$$y = (\cdots ((a_n x + a_{n-1}) x + a_{n-2}) x) \cdots) x + a_0$$

曲綫上相鄰兩點之間以直綫來趨近，而超過範圍的點則被省略。

建議練習

1. 在畫出一多項式後，容許使用者選擇讓此多項式保留在螢幕上而以相同的兩軸再畫出另一多項式。
2. 容許使用者選擇是否要畫出兩軸。
3. 在兩多項式之間以虛綫填滿之。
4. 改變輸入要求以容許使用者能輸入一多項式資料或是僅就畫出的多項式中某幾項資料做改變。
5. 改變程式以畫出 x 表爲 y 的多項式圖形。

POLYPLOT 程式　注意：如前面所述，輸入時省略 REM 敍述。

```
100  REM ••• POLYPLOT •••
110  LOMEM: 16384: REM ••• PROTECT MEMORY •••
120  DIM C0(10),C1(10),C2(10),AX(10),BY(10),D(10): REM ••• FOR QUADRATICS •••
130  DIM H0(10),H1(10),H2(10),H3(10),EX(10),FY(10),G(10): REM ••• FOR CUBICS •••
140  DIM JX(10),KY(10),GG(10),CC(100): REM ••• FOR HIGHER DEGREE POLYNOMIALS •••
150  DIM CD(10): REM ••• FOR POLYNOMIAL BEING PLOTTED •••
160  READ IM(1): REM ••• NO. OF PRESET QUADRATICS •••
170  FOR IQ=1 TO IM(1): REM ••• STORE THE DATA FOR THEM •••
```

```
180  READ C0(IQ),C1(IQ),C2(IQ),AX(IQ),BY(IQ),D(IQ)
190  NEXT IQ
200  REM *** DATA FOR QUADRATICS ***
210  DATA 5,3,2,.1,12,20,1,1,-2,-1,2,10,.1,-4,.1,.16,2.5,1.5,.1
220  DATA -3,-18,-20,.25,.5,.01,3,.2,.4,1.25,2.5,.05
230  READ IM(2): REM *** NO. OF PRESET CUBICS ***
240  FOR IQ=1 TO IM(2): REM *** STORE THE DATA FOR THEM ***
250  READ H0(IQ),H1(IQ),H2(IQ),H3(IQ),EX(IQ),FY(IQ),G(IQ)
260  NEXT IQ
270  REM *** DATA FOR CUBICS ***
280  DATA 3,1.5,2,-.06,-.05,2.5,2.5,.1,-5,-6,3,17,.25,2.5,.1
290  DATA 0,-3,1,.5,1,2.5,.1
300  READ IM(3): REM *** NO. OF PRESET HIGHER DEGREE POLYNOMIALS ***
310  IR=0
320  FOR IQ=1 TO IM(3): REM *** STORE THE DATA FOR THEM ***
330  READ DG%:IR=IR+1:CC(IR)=DG%: REM *** STORE THE DEGREE ***
340  FOR IS=0 TO DG%: REM *** STORE THE COEFFICIENTS ***
350  IR=IR+1: READ CC(IR)
360  NEXT IS
370  READ JX(IQ),KY(IQ),GG(IQ): REM *** STORE THE RANGES AND INTERVALS***
380  NEXT IQ
390  REM *** DATA FOR HIGHER DEGREE POLYNOMIALS ***
400  DATA 3,5,14,.16,.4,.9,-.01,-.01,2.5,25,.1
410  DATA 6,30,2.25,1.75,-.45,-.1,.007,.001,3,25,.1
420  DATA 6,-6,5.8,25,-83,-137,166,217,.25,5,.01
430  REM *** INITIALIZE THE CONSTANTS ***
440  WD$(0)="STRAIGHT LINE"
450  WD$(1)="QUADRATIC":WD$(2)="CUBIC":WD$(3)="POLYNOMIAL"
460  W=5:H=6:HH=INT(H/2)
470  HGR: HCOLOR=3: REM *** HI-RES GRAPHICS ***
480  VTAB 21: CALL -958: REM *** START AND CLEAR TEXT AREA ***
490  PRINT "TYPE 1 FOR SAMPLE POLYNOMIALS,"
500  PRINT " 2 TO SPECIFY A POLYNOMIAL,"
510  VTAB 23: CALL -958: REM *** START AND CLEAR THE LINE ***
520  INPUT " 3 TO QUIT: ";OP
530  ON OP GOTO 550,830,3000: REM *** BRANCH ON THE RESPONSE ***
540  PRINT "INVALID OPTION": GOTO 510
550  VTAB 21: CALL -958: REM *** START AND CLEAR TEXT AREA ***
560  PRINT "TYPE 1 FOR QUADRATICS, 2 FOR CUBICS,"
```

```
INPUT " 3 FOR OTHERS, 4 FOR MAIN OPTIONS: ";OQ
IF OQ=4 THEN 490: REM *** GO BACK TO MAIN MENU ***
IF OQ>0 AND OQ<4 THEN 620
PRINT "INVALID REQUEST": GOTO 550
REM *** USE PRESET DATA ***
IQ=0: REM *** INITIALIZE THE POINTER ***
IF IQ>=IM(OQ) THEN IQ=0: REM *** RESTART THE CYCLE IF IT HAS ENDED ***
IQ=IQ+1: REM *** MOVE TO THE NEXT POLYNOMIAL ***
ON OQ GOTO 660,700,740: REM *** BRANCH ON DEGREE ***
DG%=2: REM *** QUADRATIC-COPY THE PARAMETERS USED IN THE PLOTTING ***
CD(0)=C0(IQ):CD(1)=C1(IQ):CD(2)=C2(IQ)
XD=AX(IQ):YD=BY(IQ):DP=D(IQ)
GOTO 1110: REM *** PROCEED TO PLOT ***
DG%=3: REM *** CUBIC-COPY THE PARAMETERS USED IN PLOTTING ***
CD(0)=H0(IQ):CD(1)=H1(IQ):CD(2)=H2(IQ):CD(3)=H3(IQ)
XD=EX(IQ):YD=FY(IQ):DP=G(IQ)
GOTO 1110: REM *** PROCEED TO PLOT ***
IF IQ=1 THEN IR=0: REM *** GENERAL DEGREE-RESET POINTER AT START OF CYCLE ***
IR=IR+1
DG%=CC(IR): REM *** COPY THE DEGREE ***
FOR IS=0 TO DG%: REM ** COPY THE COEFFICIENTS ***
IR=IR+1:CD(IS)=CC(IR)
NEXT IS
XD=JX(IQ):YD=KY(IQ):DP=GG(IQ): REM *** COPY THE RANGE AND INTERVAL ***
GOTO 1110: REM *** PROCEED TO PLOT ***
REM *** POLYNOMIAL SPECS WILL BE TYPED AS INPUT ***
HGR: HCOLOR=3: VTAB 21: CALL -958: REM *** GRAPHICS & TEXT AREA CLEARED ***
INPUT "DEGREE: ";DG%
OQ=0: IF DG%<2 THEN 890: REM *** SET SWITCH FOR DESCRIPTIVE WORD ***
IF DG%=2 THEN OQ=1: GOTO 890
IF DG%=3 THEN OQ=2: GOTO 890
OQ=3
FOR TN=0 TO DG%: REM *** PROMPT FOR COEFFICIENTS ***
VTAB 22: CALL -958
IF TN>0 THEN 930
INPUT "CONSTANT=";CD(0): GOTO 970
IF TN>1 THEN 950
INPUT "COEFFT OF X=";CD(1): GOTO 970
PRINT "COEFFT OF X TO POWER ";TN;"=";
INPUT "";CD(TN)
```

```
970  NEXT TN: REM *** END OF COEFFICIENT INPUT CYCLE ***
980  VTAB 22: CALL -958: REM *** PROMPT FOR FURTHER DATA ***
990  INPUT "TICK MARK INTERVALS: ";XD,YD
1000 INPUT "X INTERVAL FOR PLOTTING: ";DP
1100 REM *** PLOT A POLYNOMIAL ***
1110 XL=0:YL=154: REM *** POSITION WHERE EQUATION WILL BE DISPLAYED ***
1120 HGR: HCOLOR=3
1130 VTAB 21: CALL -958: PRINT : REM *** CLEAR THE LINE ***
1140 FLASH : HTAB 10: PRINT "PLOTTING A ";WD$(OQ): NORMAL
1150 GOSUB 5390: REM *** DISPLAY Y ***
1160 GOSUB 5370: REM *** DISPLAY = ***
1170 SW%=0: REM *** SWITCH TO SUPRESS LEADING + SIGN ***
1180 XB%=XL: REM *** SET THE BACKTRACK POSITION ***
1190 FOR NT=0 TO DG%: REM *** CYCLE THROUGH COEFFICIENTS ***
1200 IF XL+3*W<278 THEN 1220: REM *** BRANCH IF THERE IS ENOUGH ROOM ***
1210 GOSUB 4800: GOTO 1600: REM *** ERASE ANY OF THE TERM DRAWN ALREADY ***
1220 XB%=XL: REM *** SET THE BACKTRACK POSITION ***
1230 IF CD(NT)=0 THEN 1540: REM *** BRANCH IF THE COEFFICIENT IS ZERO ***
1240 IF SW%<>0 AND CD(NT)>0 THEN  GOSUB 5350: REM *** DRAW A + ***
1250 IF CD(NT)<0 THEN  GOSUB 5330: REM *** DRAW A - ***
1260 ST$=STR$(ABS(CD(NT))):REM *** PREPARE TO DRAW THE COEFFICIENT ***
1270 IF NT>0 AND ABS(CD(NT))=1 THEN 1480: REM *** BRANCH IF COEFFT. IS 1 OR -1
1280 FOR I=1 TO LEN(ST$)
1290 IF MID$(ST$,I,1)="E" THEN 1320: REM *** BRANCH IF FLOATING POINT ***
1300 NEXT I
1310 GOTO 1400: REM *** BRANCH IF IT IS IN FIXED POINT ***
1320 EX=VAL(MID$(ST$,I+1)): REM *** EXTRACT EXPONENT OF FLOATING POINT NO. ***
1330 IF EX>=0 THEN 1400: REM *** BRANCH IF THE EXPONENT IS POSITIVE ***
1340 MA=INT(VAL(LEFT$(ST$,I-1))+.5): REM *** EXTRACT THE MANTISSA ***
1350 ST$=".": REM *** PROVIDE A PERIOD ***
1360 FOR II=-2 TO EX STEP -1: REM *** APPEND THE NECESSARY NO. OF ZEROS ***
1370 ST$=ST$+"0"
1380 NEXT II
1390 ST$=ST$+STR$(MA): REM *** APPEND THE SIGNIFICANT DIGITS ***
1400 GOSUB 4500: REM *** DRAW THE STRING ***
1480 IF NT>0 THEN  GOSUB 5420: REM *** DRAW THE X WHEN NEEDED ***
1490 IF NT<2 THEN 1530: REM *** BRANCH IF THE EXPONENT IS NOT TO BE DRAWN ***
1500 YL=YL-HH:ST$=STR$(NT): REM *** RAISE THE PLOTTING POSITION ***
1510 GOSUB 4500:REM *** DRAW THE EXPONENT ***
```

```
1520 YL=YL+HH: REM *** RESTORE THE VERTICAL PLOTTING POSITION ***
1530 SW%=1: REM *** SHOWS THAT A TERM HAS BEEN DRAWN ***
1540 NEXT NT: REM *** END OF CYCLE TO DRAW SUCCESSIVE TERMS ***
1600 GOSUB 4000: REM *** BRANCH TO PLOT AND LABEL THE AXES ***
1610 SX=24/XD:SY=24/YD: REM *** FACTORS TO GO FROM USER TO SCREEN COORDS ***
1620 FOR XU=-4*XD TO 4*XD STEP DP: REM *** CYCLE TO COMPUTE SUCCESSIVE VALUES **
1630 YU=CD(DG%): REM *** START OF POLYNOMIAL EVALUATION ***
1640 IF DG%=0 THEN YU=CD(0): GOTO 1680: REM *** DEGREE IS ZERO ***
1650 FOR TN=DG%-1 TO 0 STEP -1: REM *** CYCLE TO EVALUATE A POLYNOMIAL ***
1660 YU=YU*XU+CD(TN)
1670 NEXT TN
1680 XN=140+SX*XU:YN=72-SY*YU: REM *** CONVERT TO SCREEN COORDS ***
1690 IF XU=-4*XD THEN 1720: REM *** BRANCH AT START OF PLOT ***
1700 IF YN<0 OR 144<YN OR YP<0 OR 144<YP THEN 1720: REM ***OUT OF RANGE ***
1710 HPLOT XP,YP TO XN,YN: REM *** DRAW FROM THE PREVIOUS COMPUTED POINT ***
1720 XP=XN:YP=YN: REM *** SET THE "PREVIOUS COORDS" FOR NEXT CYCLE ***
1730 NEXT XU: REM *** END OF CYCLE TO COMPUTE SUCCESSIVE VALUES ***
1740 VTAB 21: CALL -958: REM *** PREPARE TO PROMPT FOR NEXT ACTION ***
1750 IF OP<>1 THEN 1810: REM *** BRANCH IF USER IS SPECIFYING POLYNOMIALS ***
1760 VTAB 22: PRINT "TYPE 1 FOR ANOTHER ";WD$(OQ)
1770 INPUT " 2 TO RETURN TO THE MAIN MENU: ";OP%
1780 ON OP% GOTO 630,470
1790 PRINT "INVALID REQUEST-PRESS A KEY"
1800 GET A$: GOTO 1740
1810 VTAB 22: PRINT "TYPE 1 TO SPECIFY ANOTHER POLYNOMIAL,"
1820 INPUT " 2 TO RETURN TO TO THE MAIN MENU: ";OP%
1830 ON OP% GOTO 830,470
1840 PRINT "INVALID REQUEST-PRESS A KEY"
1850 GET A$: GOTO 1810
3000 HOME : END
4000 REM *** PLOT AND LABEL THE AXES ***
4010 HPLOT 42,72 TO 238,72: REM *** X AXIS ***
4020 HPLOT 140,0 TO 140,144: REM *** Y AXIS ***
4030 XU=-4*XD:XS=40: REM *** PREPARE TO DRAW TICK MARKS ON X AXIS ***
4040 FOR N=-4 TO 4
4050 IF N=0 THEN 4080
4060 HPLOT XS,72 TO XS,67
4070 ST$=STR$(XU):XL=XS+2:YL=72+H: GOSUB 4500: REM *** DRAW THE LABEL ***
4080 XS=XS+25:XU=XU+XD
4090 NEXT N
```

```
4100  YU=-3*YD:YS=144: REM *** PREPARE TO DRAW TICK MARKS ON Y AXIS ***
4110  FOR N=-3 TO 3
4120  IF N=0 THEN 4150
4130  HPLOT 140,YS TO 145,YS
4140  ST$=STR$(YU):XL=138-W*LEN(ST$):YL=YS: GOSUB 4500
4150  YU=YU+YD:YS=YS-24
4160  NEXT N
4170  RETURN
4500  REM *** DISPLAY A STRING ***
4510  FOR I=1 TO LEN(ST$): REM *** CYCLE THROUGH SUCCESSIVE
      CHARACTERS ***
4520  IF XL+W>=279 THEN GOSUB 4800: RETURN : REM *** BRANCH IF
      TOO LONG ***
4530  CH$=MID$(ST$,I,1): REM *** EXTRACT A CHARACTER ***
4540  IF CH$="0" THEN  GOSUB 5010
4550  IF CH$="1" THEN  GOSUB 5030
4560  IF CH$="2" THEN  GOSUB 5060
4570  IF CH$="3" THEN  GOSUB 5100
4580  IF CH$="4" THEN  GOSUB 5140
4590  IF CH$="5" THEN  GOSUB 5170
4600  IF CH$="6" THEN  GOSUB 5200
4610  IF CH$="7" THEN  GOSUB 5220
4620  IF CH$="8" THEN  GOSUB 5240
4630  IF CH$="9" THEN  GOSUB 5280
4640  IF CH$=" " THEN  GOSUB 5310
4650  IF CH$="-" THEN  GOSUB 5330
4660  IF CH$="E" THEN  GOSUB 5440
4750  NEXT I
4760  RETURN
4800  HCOLOR=0: REM *** ERASE FROM BACKTRACK TO CURRENT X POSITION ***
4810  FOR YY=154-HH TO 154+H
4820  HPLOT XB%,YY TO XL,YY
4830  NEXT YY
4840  XL=XB%: HCOLOR=3
4850  GOSUB 5100: GOSUB 5100: GOSUB 5100
4860  RETURN
5000  REM *** PLOT 0 THROUGH 9 ***
5010  HPLOT XL,YL TO XL+2,YL TO XL+2,YL+4 TO XL,YL+4 TO XL,YL
5020  XL=XL+W: RETURN
5030  HPLOT XL,YL+1 TO XL+1,YL TO XL+1,YL+4
5040  HPLOT XL,YL+4 TO XL+2,YL+4
5050  XL=XL+W: RETURN
```

```
5060 HPLOT XL,YL TO XL+1,YL
5070 HPLOT XL+2,YL+1 TO XL,YL+3
5080 HPLOT XL,YL+4 TO XL+2,YL+4
5090 XL=XL+W: RETURN
5100 HPLOT XL,YL TO XL+1,YL: HPLOT XL+2,YL+1
5110 HPLOT XL+1,YL+2: HPLOT XL+2,YL+3
5120 HPLOT XL,YL+4 TO XL+2,YL+4
5130 XL=XL+W: RETURN
5140 HPLOT XL,YL TO XL,YL+2 TO XL+2,YL+2
5150 HPLOT XL+2,YL TO XL+2,YL+4
5160 XL=XL+W: RETURN
5170 HPLOT XL+2,YL TO XL,YL TO XL,YL+2 TO XL+1,YL+2
5180 HPLOT XL,YL+4 TO XL+1,YL+4: HPLOT XL+2,YL+3
5190 XL=XL+W: RETURN
5200 HPLOT XL,YL TO XL,YL+4 TO XL+2,YL+4 TO XL+2,YL+2 TO XL,YL+2
5210 XL=XL+W: RETURN
5220 HPLOT XL,YL TO XL+2,YL TO XL,YL+4
5230 XL=XL+W: RETURN
5240 HPLOT XL+1,YL TO XL+2,YL TO XL+2,YL+1
5250 HPLOT XL,YL+1: HPLOT XL+1,YL+2: HPLOT XL,YL+4
5260 HPLOT XL+1,YL+4 TO XL+2,YL+4 TO XL+2,YL+3
5270 XL=XL+W: RETURN
5280 HPLOT XL,YL+4 TO XL+2,YL+4 TO XL+2,YL TO XL,YL TO XL,YL+2
     TO XL+2,YL+2
5290 XL=XL+W: RETURN
5300 REM *** PLOT .-+=YXE ***
5310 HPLOT XL+2,YL+4
5320 XL=XL+W: RETURN
5330 HPLOT XL,YL+2 TO XL+2,YL+2
5340 XL=XL+W: RETURN
5350 HPLOT XL,YL+2 TO XL+2,YL+2: HPLOT XL+1,YL+1 TO XL+1,YL+3
5360 XL=XL+W: RETURN
5370 HPLOT XL+1,YL+1 TO XL+3,YL+1: HPLOT XL+1,YL+3 TO XL+3,YL+3
5380 XL=XL+W: RETURN
5390 HPLOT XL,YL TO XL+2,YL+2 TO XL+4,YL
5400 HPLOT XL+2,YL+4 TO XL+2,YL+2
5410 XL=XL+W: RETURN
5420 HPLOT XL,YL+4 TO XL+4,YL: HPLOT XL,YL TO XL+4,YL+4
```

```
5430  XL=XL+W+2: RETURN
5440  HPLOT XL+2,YL TO XL,YL TO XL,YL+4 TO XL+2,YL+4: HPLOT XL,YL+2 TO
      XL+2,YL+2
5450  XL=XL+W+1: RETURN
```

6—3 用點來決定多項式曲線

正如兩點之間可連成一直綫一樣，對於不同 x 值的三點可決定一二次曲綫，同理對任何四點可決定一三次曲綫，五點可決定一四次曲綫…等等，但條件是所有點的 x 值不可相同。

本節的POLYFIT 程式要求輸入 x 及 y 軸範圍，點數及其相對座標，然後畫出兩軸和各點，再自動的畫一個適當的多項式以經過這些點，同時在螢幕底部顯示出此多項式之方程式（圖 6-9 ）。此程式接著暫停以等待下一次輸入。

此程式對於太靠近的兩點相當敏感，此因程式試圖將兩個很接近零的數相除而其結果不很正確，所以最好不要輸入太靠近的點座標。同時如果程式計算各係數所需的時間與點數的平方成正比，而且其精確度與點數成反比。

此程式裡用到大量的數學運算，其中最主要的部份就是在高中學過的解幾元一次聯立方程式組。你可能自己解過三元一次聯立方程組的問題，但對高次而言，現在已經很少有人眞正動手去算它了。要將一個10次多項式通過不同的11點便牽涉到同時解出含11個未知數（此即各次方的係數，共11個）的11個一次聯立方程式組。此類綫性代數的問題必需用矩陣方法來解之。

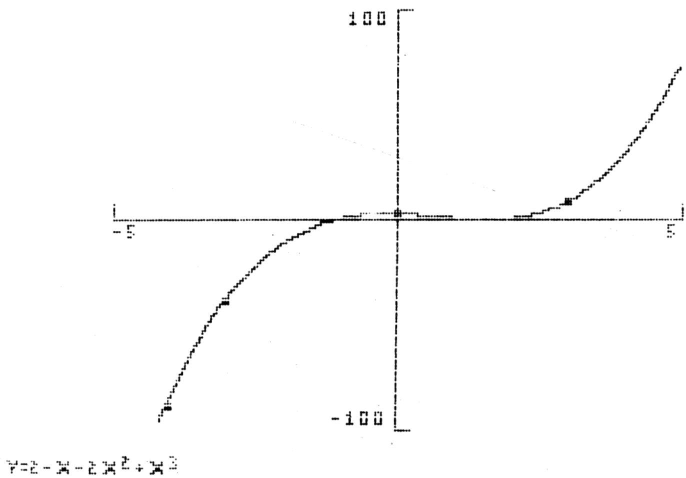


圖　6-9

POLYFIT 程式使用一種稱之爲高斯－約旦除去法（ Gauss－Jordan Elimination）的技巧來解決聯立方程式。這種方法很容易用程式寫出，但缺點是對於太接近的資料却不甚準確。近幾十年來許多數學家已在大電腦上發展出許多種更可靠的方法來解決此類問題，但對 Apple II 而言，使用這些方法還言之過早。

要求出通過 n 點的多項式實際上就是解一組 $n+1$ 個變數的聯立方程組。如果 a_0，a_1，…，a_n 代表此多項式的係數，而 x_i，y_i 代表要通過的第 i 點座標，則

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_{n-1} x_i^{n-1}$$

如果 i 由 1 取到 n 則可建立一組 n 個含未知數 a_0，a_1，a_2，…，a_n 的聯立方程式。

要對此法加以解釋，首先我們改寫方程式而以 u_k 代表未知數，而等式左邊的 y_j 改成 v_j，各係數改成 $c_{j,k}$，其中 j 與 k 分別由 1 取到 n 。

$$
\begin{aligned}
&c_{1,1}u_1+c_{1,2}u_2+\dots+c_{1,n}u_n=v_1\\
&c_{2,1}u_1+c_{2,2}u_2+\dots+c_{2,n}u_n=v_2\\
&\vdots\\
&c_{n,1}u_1+c_{n,2}u_2+\dots+c_{n,n}u_n=v_n
\end{aligned}
$$

因為有 n 個未知數，所以計算週期共需 n 次。在第一個週期裡將第一式除以 $c_{1,1}$，則係數 $c_{1,2}$ 到 $c_{1,n}$ 變成 $c_{1,2}/c_{1,1}$ 到 $c_{1,n}/c_{1,1}$，同時等式右邊變為 $v_1/c_{1,1}$ 而係數 $c_{1,1}$ 則變為 1 。第二個方程式裡的係數 $c_{2,2}$ 以 $c_{2,2}-c_{2,1}c_{1,2}$ 取代，係數 $c_{2,3}$ 以 $c_{2,3}-c_{2,1}c_{1,3}$ 取代，等等依次類推。等式的右邊以 $v_2-c_{2,1}v_1$ 取代而係數 $c_{2,1}$ 變為 0 。

對其他方程式也需做同樣的取代工作，係數 $c_{j,k}$ 由 $c_{j,k}-c_{j,1}c_{1,k}$ 取代，而所有等式右邊均變為 $v_j-c_{j,1}v_1$，第一列的每一係數則全為 0 。所以對整個方程組的第一列係數，只有第一行為 1，餘皆為 0 。

在計算的第二週期裡，由第二個方程式來扮演第一週期中第一個方程式的角色。將此方程式除以 $c_{2,2}$，餘方程式係數矩陣對角綫右邊的係數 $c_{j,k}$ 以 $c_{j,k}-c_{j,2}c_{2,k}$ 取代，而方程式右邊以 $v_j-c_{j,2}v_2$ 取代。在完成此週期後，$c_{1,1}$ 及 $c_{2,2}$ 變為 1，而前兩列的其他係數皆為 0 。

同理做第三週期時將第三個方程式除以 $c_{3,3}$，依次類推。由上面的結果可知在第 n 個週期結束時，所有的對角綫係數全變為 1，而其餘的係數則為 0，你可取 n 為 2 或 3 的簡單例子自己動手做一遍便能全盤了解如此做法的目的何在。例如用 2，5 及 1 分別代表 u_1，u_2 及 u_3，則方程式組

$$3u_1+u_2+u_3=12 \qquad u_1+2u_2-u_3=11 \qquad u_1+u_2+3u_3=10$$

最後結果還是回到2，5及1。

此程式所用的主要變數為：

W, H	字元的寬度和高度
HH	字元高度之半
XP(P)	第P點的 x 值
B(P)	第P點的 y 值
CD(NT)	多項式的第N個係數
A(R, Q)	要轉換的矩陣中第〔R，Q〕個元素
XV, XH	x 值以使用者單位計的最小及最大值
YV, YH	y 值以使用者單位計的最小及最大值
XA	y 軸的 x 螢幕座標
YA	x 軸的 y 螢幕座標
XL, YL	字串起始點的螢幕座標
ST$	欲顯示的字串
SX, SY	將使用者單位轉換為螢幕座標的因素
D	多項式要通過的點數
P	使用者用來指明各點的點數，要轉換的矩陣行數，同時也是多項式的最後係數索引
XN, YN	多項式必須通過的某點螢幕座標
Q	用來說明矩陣列數的索引變數
R	在轉換週期內的矩陣行數索引
DV	正被矩陣中低行中元素所除的對角綫元素

MY　　　　　在簡化矩陣時用做乘數的元數

OP　　　　　在求出多項式之後的功能代號

POLYFIT 程式是直接依其數學理論寫成的，求出多項式之後再用 POLYPLOT 程式的方法將此多項式畫出，同時對各軸加以分格並標明之。

建議練習

1. 修改程式而能對不同 *y* 值的數點求出以 *y* 表成 *x* 的多項式。

PLOYFIT 程式

注意：輸入前需如前一節所述將 REM 敍述刪去。

```
100  REM *** POLYFIT-FITS POLYNOMIALS ***
110  LOMEM: 16384: REM *** PROTECTS MEMORY ***
120  XI=.1:W=5:H=4:HH=2: REM *** PLOTTING INTERVAL & FONT DIMENSIONS ***
130  DIM CD(10),XP(10),A(10,10),B(10): REM *** ARRAYS FOR COMPUTATION **
140  TEXT : HOME : HGR: HCOLOR=3
150  POKE 34,22: HOME : REM *** LIMITS TEXT TO TWO BOTTOM LINES ***
160  INPUT "X LIMITS: ";XV,XH
170  INPUT "Y LIMITS: ";YV,YH
180  REM *** PLOT THE AXES ***
190  IF XV<=0 AND XH>=0 THEN XA=40-200*XV/(XH-XV): REM *** X BOTH + & -
200  IF XV>0 OR XH<0 THEN XA=40: REM *** X ENTIRELY + OR ENTIRELY - ***
210  HPLOT XA,0 TO XA,144: REM *** PLOT THE Y AXIS ***
220  IF YV<=0 AND YH>=0 THEN YA=144*YH/(YH-YV): REM *** Y BOTH + & - ***
230  IF YV>0 OR YH<0 THEN YA=144: REM *** Y ENTIRELY + OR ENTIRELY - ***
235  IF YA<10 THEN YA=144: REM *** ALLOW SPACE FOR TICK MARKS ***
240  HPLOT 40,YA TO 240,YA: REM *** PLOT THE X AXIS ***
250  HPLOT XA,0 TO XA+5,0: REM *** PLOT TICK MARK AT TOP OF Y AXIS ***
260  ST$=STR$(YH): REM *** LABEL IT ***
270  XL=XA-W*LEN(ST$)-2:YL=1: GOSUB 4500
280  HPLOT XA,144 TO XA+5,144: REM *** PLOT TICK MARK AT BOTTOM OF Y AXI
290  ST$=STR$(YV): REM *** LABEL IT ***
300  XL=XA-W*LEN(ST$)-2:YL=142-H: GOSUB 4500
```

```
310  HPLOT 40,YA TO 40,YA-5: REM *** PLOT TICK MARK AT LEFT OF X AXIS ***
320  ST$=STR$(XV): REM *** LABEL IT ***
330  XL=40:YL=YA+2: GOSUB 4500
340  HPLOT 240,YA TO 240,YA-5: REM *** PLOT TICK MARK AT RIGHT OF IT ***
350  ST$=STR$(XH): REM *** LABEL IT ***
360  XL=240-W*LEN(ST$):YL=YA+2: GOSUB 4500
370  SX=200/(XH-XV):SY=144/(YH-YV): REM *** SCALING FACTOR FOR COORDS ***
380  HOME : INPUT "NUMBER OF POINTS: ",D
390  IF D>1 AND D<11 THEN 420: REM *** BRANCH IF NUMBER OF POINTS IS VALID ***
400  FLASH : PRINT "INVALID NUMBER-PRESS A KEY": NORMAL : GET A$: GOTO 380
410  POKE 34,23: REM *** PROTECT LINE 23 ON THE SCREEN ***
420  FOR P=1 TO D: REM *** CYCLE TO STORE THE SUCCESSIVE POINTS ***
430  HOME : INPUT "COORDINATES: ";XP(P),B(P)
440  A(P,1)=1
450  FOR Q=2 TO D: REM *** STORE THE POWERS OF X IN THE A MAT  X ***
460  A(P,Q)=XP(P)*A(P,Q-1)
470  NEXT Q
480  XN=40+SX*(XP(P)-XV): REM *** FORM THE SCREEN COORDS FOR NEW POINT ***
490  YN=144-SY*(B(P)-YV)
500  IF 40>XN OR 240<XN OR 0>YN OR 144<YN THEN 540: REM *** NOT PLOTTABLE ***
510  HPLOT XN,YN: REM *** PLOT THE POINT AND A SQUARE AROUND IT IF POSSIBLE ***
520  IF YN>0 AND XN>40 AND XN<240 THEN  HPLOT XN-1,YN TO XN-1,YN-1 TO XN+1,YN-1 TO
     XN+1,YN
530  IF YN<144 AND XN>40 AND XN<240 THEN HPLOT XN-1,YN TO XN-1,YN+1 TO XN+1,YN+1 T
     O XN+1,YN
540  HOME : NEXT P: REM *** END OF CYCLE TO STORE THE SUCCESSIVE POINTS ***
550  REM *** SOLVE THE EQUATIONS ***
560  POKE 34,22:HOME:HTAB 10:FLASH:PRINT "COMPUTING THE COEFFICIENTS":NORMAL
570  FOR P=1 TO D: REM *** PERFORM THE MAIN ITERATION D TIMES ***
580  IF A(P,P)=1 THEN 650: REM *** BRANCH IF DIAGONAL ELEMENT IS 1 ALREADY ***
590  DV=A(P,P):A(P,P)=1: REM *** DIVIDE THE ROWS BELOW IT BY IT ***
600  IF P=D THEN 640: REM *** BRANCH IF AT THE BOTTOM ROW ALREADY ***
610  FOR Q=P+1 TO D: REM *** START OF CYCLE TO DIVIDE LOWER ROWS ***
620  A(P,Q)=A(P,Q)/DV
630  NEXT Q: REM *** END OF CYCLE TO DIVIDE THE LOWER ROWS ***
640  B(P)=B(P)/DV: REM *** DIVIDE THE P-TH ELEMENT IN THE RESULT VECTOR ***
650  FOR R=1 TO D: REM *** START OF OUTER CYCLE TO REDUCE ELEMENTS ***
660  IF R=P THEN 740: REM *** BYPASS ACTIONS FOR P-TH ROW ***
670  MY=A(R,P): REM *** SELECT ELEMENT FROM R-TH ROW ***
680  A(R,P)=0
```

```
690  IF P=D THEN 730
700  FOR Q=P+1 TO D: REM *** START OF INNER CYCLE FOR RIGHT OF DIAGONAL ***
710  A(R,Q)=A(R,Q)-MY*A(P,Q)
720  NEXT Q: REM *** END OF INNER CYCLE FOR RIGHT OF DIAGONAL ELEMENTS ***
730  B(R)=B(R)-MY*B(P): REM *** REDUCE THE R-TH ELEMENT IN THE RESULT VECTOR ***
740  NEXT R: REM *** END OF OUTER CYCLE TO REDUCE ELEMENTS ***
750  NEXT P: REM *** END OF CYCLE FOR MAIN ITERATION ***
760  FOR TN=0 TO D-1: REM *** TRANSFER RESULTS FOR CONSISTENCY WITH POLYPLOT ***
770  CD(TN)=B(TN+1)
780  NEXT TN
790  DG%=D-1
800  HOME : REM *** CLEAR THE TEXT AREA ***
1100 REM *** PLOT THE POLYNOMIAL-BEGIN BY DRAWING THE EQUATION (SEE POLYPLOT) ***
1110 XL=0:YL=155
1150 GOSUB 5390: REM *** DISPLAY Y ***
1160 GOSUB 5370: REM *** DISPLAY = ***
1170 SW%=0
1180 XB%=XL
1190 FOR NT=0 TO DG%
1200 IF XL+3*W<278 THEN 1220
1210 GOSUB 4700: GOTO 1600
1220 XB%=XL
1230 IF CD(NT)=0 THEN 1540
1240 IF SW%<>0 AND CD(NT)>0 THEN  GOSUB 5350: REM *** + ***
1250 IF CD(NT)<0 THEN  GOSUB 5330: REM *** - ***
1260 ST$=STR$(ABS(CD(NT)))
1270 IF NT>0 AND ABS(CD(NT))=1 THEN 1480
1280 FOR I=1 TO LEN(ST$)
1290 IF MID$(ST$,I,1)="E" THEN 1320
1300 NEXT I
1310 GOTO 1410
1320 EX=VAL(MID$(ST$,I+1))
1330 IF EX>=0 THEN 1470
1340 MA=INT(VAL(LEFT$(ST$,I-1))+.5)
1350 ST$="."
1360 FOR II=-2 TO EX STEP -1
1370 ST$=ST$+"0"
1380 NEXT II
1390 ST$=ST$+STR$(MA)
1400 GOTO 1470
```

```
1410 IF LEN(ST$)<6 THEN 1470: REM *** BRANCH IF # LESS THAN 6 DIGITS ***
1420 FOR I=1 TO LEN(ST$): REM *** SCAN FOR DECIMAL POINT ***
1430 IF MID$(ST$,I,1)="." THEN 1460
1440 NEXT I
1450 GOTO 1470
1460 IF I+3<LEN(ST$) THEN ST$=LEFT$(ST$,I+3): REM *** TRUNCATE IF POSSIBLE ***
1470 GOSUB 4500: REM *** COEFFICIENT ***
1480 IF NT>0 THEN  GOSUB 5420: REM *** X ***
1490 IF NT<2 THEN 1530
1500 YL=YL-HH:ST$=STR$(NT)
1510 GOSUB 4500: REM *** EXPONENT ***
1520 YL=YL+HH
1530 SW%=1
1540 NEXT NT
1600 HOME : PRINT : FLASH : HTAB 10: PRINT "PLOTTING THE POLYNOMIAL": NORMAL
1610 FOR XU=XV TO XH STEP XI: REM *** PLOTTING CYCLE ***
1620 YU=CD(DG%). REM *** BEGIN THE EVALUATION ***
1630 IF DG%=0 THEN 1670
1640 FOR TN=DG%-1 TO 0 STEP -1: REM *** CYCLE FOR SUCCESSIVE COEFFICIENTS ***
1650 YU=YU*XU+CD(TN)
1660 NEXT TN
1670 XN=40+SX*(XU-XV):YN=144-SY*(YU-YV): REM *** SCREEN COORDINATES ***
1680 IF XU=XV THEN 1710
1690 IF YN<0 OR 144<YN OR YP<0 OR 144<YP THEN 1710
1700 HPLOT XP,YP TO XN,YN: REM *** DRAW FROM PREVIOUS POINT ***
1710 XP=XN:YP=YN
1720 NEXT XU
1721 REM *** PROMPT FOR FURTHER ACTION ***
1730 POKE 34,21: HOME : PRINT "TYPE 1 TO LIST COEFFICIENTS,"
1740 PRINT "      2 TO FIT ANOTHER POLYNOMIAL,"
1750 INPUT " 3 TO QUIT: ";OP
1760 ON OP GOTO 1780,140,1840
1770 FLASH : PRINT "INVALID REQUEST": NORMAL : GOTO 1730
1780 POKE 34,0: TEXT : HOME
1790 PRINT "POWER OF X","COEFFICIENT": PRINT
1800 FOR TN=0 TO DG%: REM *** DISPLAY THE COEFFICIENTS ***
1810 PRINT TN,CD(TN)
1820 NEXT TN
1830 GOTO 1730
1840 POKE 34,0: TEXT : HOME : END : REM *** QUIT ***
```

```
4500 REM *** DISPLAY A STRING ***
4510 FOR I=1 TO LEN(ST$)
4530 CH$=MID$(ST$,I,1)
4540 IF CH$="0" THEN  GOSUB 5010
4550 IF CH$="1" THEN  GOSUB 5030
4560 IF CH$="2" THEN  GOSUB 5060
4570 IF CH$="3" THEN  GOSUB 5100
4580 IF CH$="4" THEN  GOSUB 5140
4590 IF CH$="5" THEN  GOSUB 5170
4600 IF CH$="6" THEN  GOSUB 5200
4610 IF CH$="7" THEN  GOSUB 5220
4620 IF CH$="8" THEN  GOSUB 5240
4630 IF CH$="9" THEN  GOSUB 5280
4640 IF CH$=" " THEN  GOSUB 5310
4650 IF CH$="-" THEN  GOSUB 5330
4660 IF CH$="E" THEN  GOSUB 5440
4750 NEXT I
4760 RETURN
5000 REM *** PLOT 0 THROUGH 9 ***
5010 HPLOT XL,YL TO XL+2,YL TO XL+2,YL+4 TO XL,YL+4 TO XL,YL
5020 XL=XL+W: RETURN
5030 HPLOT XL,YL+1 TO XL+1,YL TO XL+1,YL+4
5040 HPLOT XL,YL+4 TO XL+2,YL+4
5050 XL=XL+W: RETURN
5060 HPLOT XL,YL TO XL+1,YL
5070 HPLOT XL+2,YL+1 TO XL,YL+3
5080 HPLOT XL,YL+4 TO XL+2,YL+4
5090 XL=XL+W: RETURN
5100 HPLOT XL,YL TO XL+1,YL: HPLOT XL+2,YL+1
5110 HPLOT XL+1,YL+2: HPLOT XL+2,YL+3
5120 HPLOT XL,YL+4 TO XL+2,YL+4
5130 XL=XL+W: RETURN
5140 HPLOT XL,YL TO XL,YL+2 TO XL+2,YL+2
5150 HPLOT XL+2,YL TO XL+2,YL+4
5160 XL=XL+W: RETURN
5170 HPLOT XL+2,YL TO XL,YL TO XL,YL+2 TO XL+1,YL+2
5180 HPLOT XL,YL+4 TO XL+I,YL+4: HPLOT XL+2,YL+3
5190 XL=XL+W: RETURN
5200 HPLOT XL,YL TO XL,YL+4 TO XL+2,YL+4 TO XL+2,YL+2 TO XL,YL+2
```

```
XL=XL+W: RETURN
HPLOT XL,YL TO XL+2,YL TO XL,YL+4
XL=XL+W: RETURN
HPLOT XL+1,YL TO XL+2,YL TO XL+2,YL+1
HPLOT XL,YL+1: HPLOT XL+1,YL+2: HPLOT XL,YL+4
HPLOT XL+1,YL+4 TO XL+2,YL+4 TO XL+2,YL+3
XL=XL+W: RETURN
HPLOT XL,YL+4 TO XL+2,YL+4 TO XL+2,YL TO XL,YL TO XL,YL+2 TO
XL+2,YL+2
XL=XL+W: RETURN
REM *** PLOT  -+=YXE ***
HPLOT XL+2,YL+4
XL=XL+W: RETURN
HPLOT XL,YL+2 TO XL+2,YL+2
XL=XL+W: RETURN
HPLOT XL,YL+2 TO XL+2,YL+2: HPLOT XL+1,YL+1 TO XL+1,YL+3
XL=XL+W: RETURN
HPLOT XL+1,YL+1 TO XL+3,YL+1: HPLOT XL+1,YL+3 TO XL+3,YL+3
XL=XL+W: RETURN
HPLOT XL,YL TO XL+2,YL+2 TO XL+4,YL
HPLOT XL+2,YL+4 TO XL+2,YL+2
XL=XL+W: RETURN
HPLOT XL,YL+4 TO XL+4,YL: HPLOT XL,YL TO XL+4,YL+4
XL=XL+W+2: RETURN
HPLOT XL+2,YL TO XL,YL TO XL,YL+4 TO XL+2,YL+4: HPLOT XL,YL+2
TO XL+2,YL+2
XL=XL+W+1: RETURN
```

6−4 使角平滑的曲線

在畫如臉部圖形，一輛車子或海灘上波浪等圖形時，想不想只概廓的描出幾點而讓電腦來畫出平滑的區形？圖6-10中臉部的概廓圖便是用近幾年來廣泛應用在設計汽車及飛機的一種由一位法國工程師貝其爾所發明的方法將其修飾成如圖6-11的平滑曲綫。這種方法到目前爲止仍然很昂貴，

但讀完本節之後，你便可在自己的 Apple 上實驗這種目的。

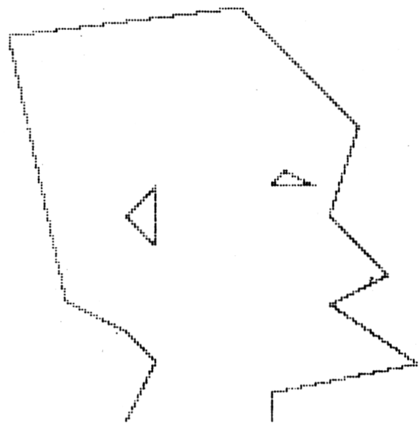


圖 6-10 〞點對點〞畫圖

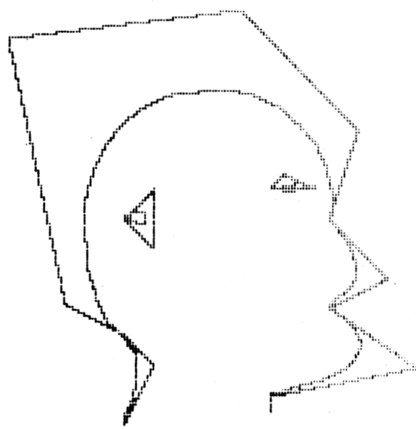


圖 6-11 上圖之平滑曲線

這個方法的基本原理是用一種稱爲〃法國曲綫〃（ French curves ，參見圖6-12 ）的曲綫來將綫段和有稜角的部份做平滑的趨近，這種稱爲貝斯爾平滑曲綫法是用一組如圖 6-13A 及 6-13B 的曲綫來對不同點數的輪廓圖畫出其平滑曲綫，BEZFUN 程式主要是將這些曲綫畫出而爲下一節的 BEZFIT 程式做平滑工作之用 。

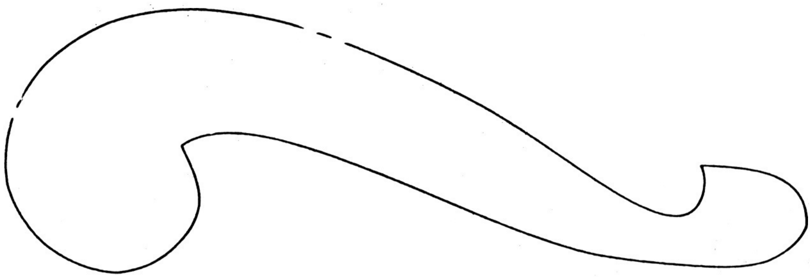


圖 6-12 〃法國曲綫〃圖

貝斯爾函數由 0 畫到 1 ，水平軸的變數我們不用 x ，而用 u 來代表以說明這些函數 。每一種函數相對於一個特別的值 ，n 的取值較輪廓圖的點數少 1 ，而 i 值爲 0 到 n 。例如只有兩點時， n 爲 1 ，而 i = 0 及 i = 1 的函數圖形僅爲直綫，此與兩點決定一綫的幾何常識不相衝突 。

$$b(1,0,u)=1-u \quad \text{and} \quad b(1,1,u)=u$$

我們用符號 b (n , i , u) 來代表一貝斯爾函數，當 n = 2 時，其相對的三個貝斯爾函數爲：

$$b(2,0,u)=(1-u)^2, \qquad b(2,1,u)=2u(1-u), \qquad b(2,2,u)=u^2$$

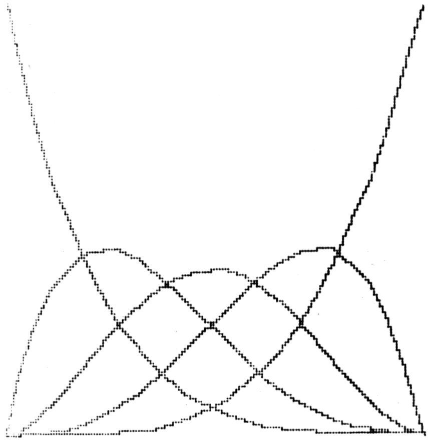


圖 6-13A N=4

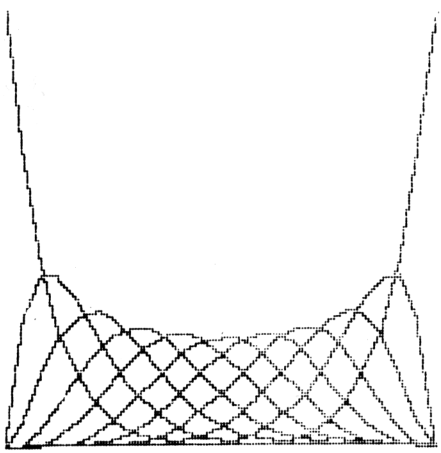


圖 6-13B N=10

而 $n = 3$ 時，相對的四個函數為：

$$b(3,0,u)=(1-u)^3,\qquad b(3,1,u)=3u(1-u)^2$$
$$b(3,2,u)=3u^2(1-u),\qquad b(3,3,u)=u^3$$

將這些函數推廣到高次 n 值時只需使用展開 $(1+x)^n$ 的二項式定理，如果以數學觀點來解釋，這些函數簡單的說就是二項式機率分佈函數，故可用來將輪廓圖平滑之。

在程式裡計算貝斯爾函數的方法是先由下列式子開始：

$$b(n,0,u)=(1-u)^n$$

並使用公式

$$b(n,i,u)=\frac{(n-i+1)}{i}\times\frac{u}{(1-u)}\times b(n,i-1,u)$$

設定 i 為 1 便可由 $b(n,0,u)$ 算出 $b(n,1,u)$ 的值，設定 i 為 2 便可由 $b(n,1,u)$ 算出 $b(n,2,u)$ 的值，依此類推，直到 $i=n$。

程式裡用到的變數為：

N	控制點數 $n+1$
I，U	相對於公式裡的 i，u
NU	畫函數時所需的點數
NP	除去邊界各點後的點數
IP	連續由 0 取到 NP，亦即當 IP = 0 時 U = 0，IP = NP 時 U = 1 …等
BN (I)	非現行 u 值的 $b(n,i,u)$ 函數值

UO　　　　曲綫上前一點的 *u* 值

BO (I)　　曲綫上前一 *u* 值的 *b* (*n* , *i* , *u*) 函數值

建議練習

1. 試對不同的N值畫出其貝斯爾函數圖。
2. 利用這些圖來做一些動態效果。

BEZFUN 程式

```
100  REM *** BEZFUN-PLOTS BEZIER FUNCTIONS ***
110  TEXT : HOME : HGR: HCOLOR=3: POKE 34,20
120  HOME : INPUT "NUMBER OF CONTROL POINTS: ";NC
130  N=NC-1: IF N<2 THEN 120
140  INPUT "NUMBER OF POINTS PER CURVE: ";NU
150  NP=NU-2: IF NP<1 THEN 120
170  DIM BO(N),BN(N)
180  BO(0)=1
190  FOR IP=1 TO NP: REM *** CYCLE THROUGH THE POINTS ***
200  U=IP/(NP+1):BN(0)=(1-U) ^ N
220  BN(I)=(N-I+1)/I*U/(1-U)*BN(I-1)
250  HPLOT 150*UO,159-150*BO(I) TO 150*U,159-150*BN(I): REM ***
     JOIN POINTS ***
260  BO(I)=BN(I)
270  NEXT I
280  UO=U
290  NEXT IP
300  FOR I=0 TO N-1: REM *** COMPLETE THE CURVES THAT END AT 1 ***
310  HPLOT 150*UO,159-150*BN(I) TO 150,159
320  NEXT I
330  HPLOT 150*UO,159-150*BN(N) TO 150,9: REM *** THIS ENDS AT 0 ***
340  END
```

6—5 BEZEIT程式

現在我們可以來用前一節所畫出的〃平滑曲綫工具〃。BEZFIT程式首先顯示主指示表：

```
TYPE 1 TO PLOT CURVE,  2 TO ERASE CURVE,
     3 TO DRAW LINE,   4 TO CLEAR SCREEN,
     5 TO ALTER CURVE, 6 TO QUIT:
```

如果你打入1，程式要求輸入輪廓圖形內的點數，這些稱之爲〃控制點〃。注意輸入各點的順序必須使其形成一個〃洞〃，也就是說你不可以在輪廓上裡外的輸入各點。請參見下面的例子及圖6-14及圖6-15。

此程式給予使用者是否顯示出控制點的選擇，在每輸入一組座標後，程式會問你是否需記住或將其改變。在所有控制點座標輸入完後，程式利用前面說過的方法計算出連接各控制點間的趨近點座標，並將此平滑曲綫畫出，在連接各中間趨近點時，程式會問你是否要將此點包括進圖形中，最後程式也會讓使用者決定是否僅畫出此曲綫或同時也將連接控制點的直綫輪廓畫出。

下面是此程式的兩個範例：

資料：

控制點數＝4

座標（0，0），（40，80），（240，80），（279，0）

所產生圖形如圖 6-14

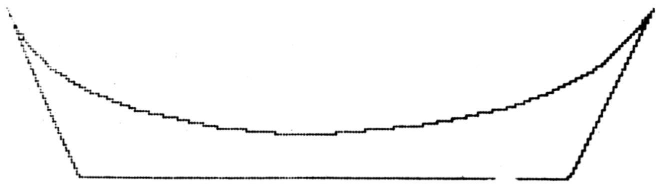


圖 6-14

控制點數＝5

座標（0，0），（120，0），（120，120），（0，120），（0，0）

產生圖形如圖 6-15

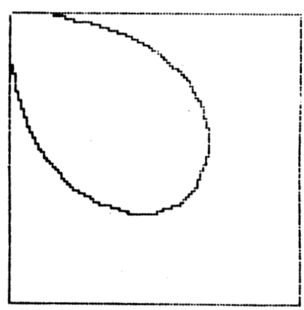


圖 6-15

在顯示出曲線後，程式再回到主指示表，你可依各功能選擇繼續做想做的事。打入 2 是將原曲線擦去，然後再接受新資料來畫圖，如果你打入 3，則程式要求你輸入一直線的兩端點座標，而在接受輸入後再將其畫出。打入 4 則清除螢幕。

如果你打 5 ，則顯示次指示表：

```
TYPE 1 (ERASE LAST CURVE), 2 (DISPLAY
COORDINATES), 3 (ALTER A POINT),
4 (DISPLAY NEW CURVE), 5 (RETURN TO
MAIN MENU):
```

此時如打入 1 則將原曲綫擦去，打入 2 則顯示區綫的控制點座標，記住這些座標除非你用選擇碼 1 將曲綫擦去否則將一直由程式記著。打入 4 使你可說明前一曲綫控制點的順序號，並顯示出座標值以接受你的改變，你可用此方法改變原曲綫的數個控制點，然後打入 4 以將新的曲綫畫出，所以本程式可用在〞交談式〞的圖形設計應用上。

貝斯爾方法的理論如下。當我們在畫平滑的曲綫時，變數 u 的取値由 0 逐漸到 1 。如果控制點數爲 9 ，則 u 的取値分別爲 0 ，0.1 ，0.2 ，0.3 ，… 1.0 。而對每一 u 値所畫出的點 x 座標爲：

$$x=x_0b(n,0,u)+x_1b(n,1,u)+\ldots+x_nb(n,n,u)$$

此處 x_0 ，x_1 ，… ，x_n爲各控制點 x 值 。b 就是前一節所述的現行貝斯爾函數值，而此點的 y 座標爲：

$$y=y_0b(n,0,u)+y_1b(n,1,u)+\ldots+y_nb(n,n,u)$$

y_0 ，y_1 ，… ，y_n爲控制點 y 值 。

因於貝斯爾函數的特性，利用上面的公式便可在控制點間求出各中間趨近點而依此畫出平滑曲綫 。如 $n=3$ ，則圖示在圖6-16 。此程序開始時 $u=0$ 。因爲 $b(3,0,u)$ 爲 1 ，而其餘 b 值皆爲 0 ，所以此點的 x ，y 座標就是控制點座標 x_0 ，y_0 。當 u 增加時，便由公式求出逐漸增加的 x ，

*y*座標。這種增加的趨勢在*u*取到*b*(3，1，*u*)的最大值時最爲明顯，所以此時這一個控制點完全掌握曲綫在這一點的特性，也就是說，我們輸入了圖形輪廓的數個控制點後，程式便用貝斯爾方法來在各控制點間取出適當的點，而兩相鄰控制點對其中間的趨近點的影響就是貝斯爾函數的用處所在。

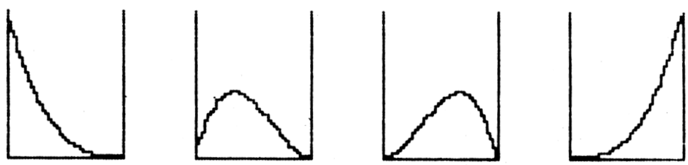


圖 6-16

此程式用到的主要變數爲：

NC	控制點數
N	控制點數減 1
I	對各控制點取 0，1，…，N 的連續值
X(I)	第 I 個控制點的 *x* 座標
Y(I)	第 I 個控制點的 *y* 座標
NP	兩控制點間的趨近點數
IP	上述各點的索引，取值爲 0，1，…到 NP＋1(0 及 NP＋1 代表兩端點)
U	隨著曲綫由 0 逐漸取到 1 的變數
XP(IP)	第 IP 點的 *x* 值
YP(IP)	第 IP 點的 *y* 值

B(I)　　對現存N,I,U的貝斯爾函數 $b(n,i,u)$ 值

此程式前先要求輸入資料，當使用者定義的一輪廓後，程式便到FOR迴路裡，對U值計算各個貝斯爾函數，欲描出的點座標XP(IP),YP(IP)則第三個迴路計算，並保存之，直到 IP 到達其範圍爲止。然後程式利用前面計算好的點座標將平滑曲綫畫出，然後顯示出指示表。

建議練習

1. 試驗(1)增加控制點數及(2)使控制點很接近時的效果。
2. 容許程式能將計算出的平滑曲綫各點座標存到檔案中以備往後的需要。
3. 在螢幕邊緣上畫出範圍邊界。
4. 利用光點位置來輸入各控制點資料（參見第 3-3 節）。
5. 利用搖捍來輸入各控制點資料。

BEZFIT 程式

```
100 REM   *** BEZFIT FOR INTERACTIVE BEZIER FITTING ***
110 DIM X(20),Y(20),B(20),XP(100),YP(100)
120 HGR: HCOLOR=3: POKE 34,20: HOME :
130 HOME : PRINT "TYPE 1 TO PLOT CURVE, 2 TO ERASE CURVE,"
140 PRINT " 3 TO DRAW LINE, 4 TO CLEAR SCREEN,"
150 INPUT " 5 TO ALTER A CURVE, 6 TO QUIT: ";OP%
160 HOME : ON OP% GOTO 200,410,430,460,510,730
170 PRINT "INVALID REQUEST: ";OP%
180 FOR DE=1 TO 1000: NEXT DE: GOTO 130
190 REM *** DEFINE A HULL ***
200 HOME : INPUT "NUMBER OF CONTROL POINTS: ";NC
210 N=NC-1
220 HOME : INPUT "DISPLAY POINTS?(Y OR N) ";Q$:Q$=LEFT$(Q$,1)
230 FOR I=0 TO N: REM *** INPUT CYCLE FOR THE CONTROL POINTS ***
240 HOME : PRINT "COORDINATES OF CONTROL POINT ";I+1;: INPUT ": ";X,Y
```

```
250  IF X<=279 AND Y<=159 AND X>=0 AND Y>=0 THEN 270
260  PRINT "***ILLEGAL VALUE***": GET A$: GOTO 240
270  X(I)=X:Y(I)=Y
280  IF Q$="N" THEN 300:REM *** BRANCH IF NOT TO PLOT ***
290  HPLOT X,Y
300  HOME : INPUT "ADEQUATE?(Y OR N) ";RQ$
310  IF RQ$<>"N" THEN 330
320  HCOLOR=0: HPLOT X,Y: HCOLOR=3: GOTO 240: REM *** IT WAS ERASED "
330  NEXT I: REM *** END OF INPUT CYCLE ***
340  HOME : INPUT "NO. OF POINTS TO BE PLOTTED (<=50): ";NU
350  NP=NU-2: IF NP>48 THEN 360
360  IF Q$<>"Y" THEN 390
370  HCOLOR=0: REM *** ERASE THE CONTROL POINTS ***
380  FOR I=0 TO N: HPLOT X(I),Y(I): NEXT I
390  HCOLOR=3: GOSUB 1010: GOTO 130
400  REM *** ERASE A CURVE ***
410  HCOLOR=0: GOTO 200
420  REM *** DRAW A STRAIGHT LINE ***
430  INPUT "END POINTS OF THE LINE? ";X1,Y1,X2,Y2
440  HPLOT X1,Y1 TO X2,Y2: GOTO 130
450  REM *** CLEAR THE SCREEN ***
460  HGR: HCOLOR=3: GOTO 130
500  REM *** ALTER A CURVE ***
510  HOME : PRINT "TYPE 1 (ERASE LAST CURVE), 2 (DISPLAY"
520  PRINT "COORDINATES), 3 (ALTER A POINT),"
530  PRINT "4 (DISPLAY NEW CURVE), 5 (RETURN TO "
540  INPUT "MAIN MENU): ";OQ%
550  HOME : ON OQ% GOTO 590,610,690,390,130
560  PRINT "INVALID REQUEST: ";OQ%
570  PRINT "PRESS ANY KEY TO TRY AGAIN": GET A$: GOTO 510
580  REM *** ERASE THE LAST CURVE **
590  HCOLOR=0: GOSUB 1010: HCOLOR=3: GOTO 510
600  REM *** DISPLAY THE COORDINATES ***
610  TEXT: HOME: INVERSE: PRINT "I.D.#","X-COORD","Y-COORD": NORMAL:
     VTAB 4
620  FOR I=0 TO N
630  PRINT I+1,X(I),Y(I)
640  NEXT I
650  PRINT "HIT ANY KEY TO CONTINUE ";: GET A$
660  POKE -16304,0: POKE -16297,0: POKE 34,20: HOME
```

```
670  GOTO 510
680  REM *** ALTER A CONTROL POINT ***
690  INPUT "ID# OF POINT: ";I:I=I-1
700  PRINT "OLD COORDS: ";X(I),Y(I)
710  INPUT "NEW COORDS: ";X(I),Y(I)
720  GOTO 510
730  TEXT : HOME : END
1000 REM *** COMPUTE BEZIER FUNCTIONS AND PLOT THE CURVE ***
1010 XP(0)=X(0):YP(0)=Y(0)
1020 IF OP%=2 THEN  HCOLOR=0
1030 FOR IP=1 TO NP: REM *** CYCLE FOR SUCCESSIVE PLOTTING POINTS ***
1040 U=IP/(NP+1):B(0)=(1-U) ^ N
1050 FOR I=1 TO N: REM *** CYCLE FOR SUCCESSIVE ORDERS ***
1060 B(I)=(N-I+1)/I*U/(1-U)*B(I-1)
1070 NEXT I: REM *** END CYCLE FOR SUCCESSIVE ORDERS ***
1080 XP(IP)=0:YP(IP)=0
1090 FOR I=0 TO N: REM *** COMPUTE THE WEIGHTED COORDINATES ***
1100 XP(IP)=XP(IP)+B(I)*X(I)
1110 YP(IP)=YP(IP)+B(I)*Y(I)
1120 NEXT I: REM *** END OF COMPUTATION OF COORDINATES ***
1130 NEXT IP: REM *** END OF CYCLE FOR SUCCESSIVE PLOTTING POINTS ***
1140 XP(NP+1)=X(N):YP(NP+1)=Y(N)
1150 HOME : INPUT "TYPE (C)URVE OR (B)OTH CURVE AND HULL: ";IN$
1160 IF IN$="C" THEN 1200
1170 FOR I=1 TO N: REM *** DRAW THE HULL ***
1180 HPLOT X(I),Y(I) TO X(I-1),Y(I-1)
1190 NEXT I
1200 FOR IP=1 TO NP+1: REM *** DRAW THE CURVE ***
1210 HPLOT XP(IP),YP(IP) TO XP(IP-1),YP(IP-1)
1220 NEXT IP
1230 HCOLOR=3
1240 RETURN
```

6－6 橢圓軌道

本節的 ORBIT 程式可用來動態顯示衛星在軌道上運行的圖形，或顯示電子繞核子運動的情形，如圖6-17所示，代表物體的小方塊在做繞行運動時本身還會做旋轉的動作。

此程式首先要求輸入運動物體數及其體積大小和運動速度，同時也要求輸入定義軌道的兩數，當軌道爲圓形時，此二數皆等於軌道半徑，當軌道爲橢圓時；此二數分別代表其長軸及短軸長。試試看輸入 5 個運動物體而其速度，大小及軌道定義值分別在 2 到12，10 到 120 及10到70之間的圖形結果。

此程式以螢幕中心爲座標系之原點，然後用

$$x=a\cos\theta,\qquad y=b\sin\theta$$

來計算物體的座標。其中 a ，b 爲定義此軌道的數值，而在動態效果時（即物體移動時）θ 值以正比於速度的量逐漸增大。當然圖6-17祇是行星軌道的趨近圖，眞正在行星運動中，太陽應位在橢圓的焦點上。（上述各幾何名詞已在高中數學教材中定義）

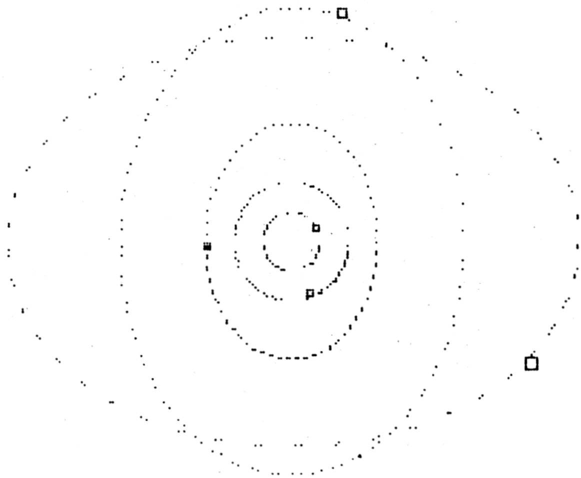


圖 6-17

本程式用到的主要變數爲：

TV (2 , 314)	程式中計算座標時對 θ 的正弦及餘弦表
NO	運動物體數
L	上述各運動物體的編號
XF (L)	第 L 個物體的水平軸長
YF (L)	第 L 個物體的垂直軸長
XN (L)	第 L 個物體的 x 座標
YN (L)	第 L 個物體的 y 座標
XO (L)	XN (L) 的前一值
YO (L)	YN (L) 的前一值
PN (L)	第 L 個物體的運動角度

RN(L)　　第L個物體的自轉角度(以 $45\pi/8$ 弳為單位)
RO(L)　　RN(L)的前一值

此程式首先存入做為代表小方塊的圖形表資料碼並計算 sin 和 cos 函數值以加快動態的效果。接著要求輸入運動物體數及其他相對的資料，然後程式執行運行軌道的週期。

主週期內對每一物體都含一內週期，在內週期裡將角速度 SP(L)加入 PN(L)裡以求出第L個運動物體的角位置，然後由 TV 列陣中取出其相對正弦及餘弦值以計算座標。同時自旋角量 RN(L)也需加上 SP(L)以計算本身的自旋，計算座標的方法前面已提過。然後用 ROT 和DRAW 指令將此物體畫在新的位置上，同時在畫下一位置時必須將原位置上的物體擦去。

建議練習

1. 容許使用者在軌道兩軸上標明名稱。
2. 使用其他各種不同的圖形來代表物體(此處為方形)。
3. 將軌道中心(太陽)放到橢圓的焦點而非原點。
4. 利用三角函數合成的公式先計算1°的 sin 及 cos 值，然後計算2°，3°…等值以減少計算時間。

ORBIT 程式

```
100  REM *** ORBIT-PLOTS ELLIPTICAL PATHS ***
110  LOMEM: 16384
120  DIM TV(2,314)
130  HGR: HCOLOR=3: POKE 34,20: HOME : PRINT "*** PLEASE WAIT-COMPUTING ***"
140  SCALE=7: ROT=0:P=0
```

```
150 FOR I=7676 TO 7682: READ V: POKE I,V: NEXT I: REM *** STORE
    SHAPE TABLE ***
160 POKE 232,252: POKE 233,29: REM *** STORE ITS BASE ADDRESS ***
170 DATA 1,0,4,0,44,62,0
180 FOR A=0 TO 6.28 STEP .02
190 TV(1,P)=COS(A):TV(2,P)=SIN(A):P=P+1
200 NEXT A
210 HOME : INPUT "HOW MANY ORBITING OBJECTS? ";NO
220 IF NO>10 THEN  PRINT "NO MORE THAN 10":NO=10
230 FOR L=1 TO NO
235 HOME : PRINT "OBJECT ";L;" OF ";NO
240 INPUT "HALF LENGTHS OF AXES=";XF(L),YF(L)
250 INPUT "SPEED AND SIZE=";SP(L),SC(L)
260 NEXT L
270 FOR L=1 TO NO
280 RN(L)=RN(L)+SP(L)
290 IF RN(L)>64 THEN RN(L)=RN(L)-64
300 PN(L)=PO(L)+SP(L)
310 IF PN(L)>314 THEN PN(L)=PN(L)-314
320 NEXT L
330 FOR L=1 TO NO
340 XN(L)=140+XF(L)*TV(1,PN(L))
350 YN(L)=80+YF(L)*TV(2,PN(L))
360 NEXT L
370 FOR L=1 TO NO
380 SCALE=SC(L): ROT=RO(L)
390 XDRAW 1 AT XO(L),YO(L)
400 ROT=RN(L)
410 XDRAW 1 AT XN(L),YN(L)
420 HPLOT XO(L),YO(L)
430 NEXT L
440 FOR L=1 TO NO
450 XO(L)=XN(L):YO(L)=YN(L)
460 RO(L)=RN(L):PO(L)=PN(L)
470 NEXT L
480 GOTO 270
```

6—7 其他圖形

本節的 ROULETTE 程式可畫出許多種前面沒有的圖形，圖 6-18A 到 6-18C 及圖 6-19A 到 6-19C 便是其中的幾個例子。

此程式首先要求選擇圖形的種類代號 E 或 H，然後要求你輸入此圖形的葉數和大小。對於圖 6-18A 到圖 6-19C 的輸入資料分別爲：

E,2,5　E,3,5　E,4,5
H,3,5　H,4,5　H,5,5

其中大小範圍應爲 2 到 8 。



圖 6-18A

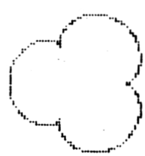


圖 6-18B

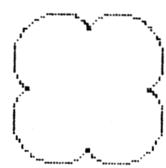


圖 6-18C



圖 6-19A



圖 6-19B



圖 6-19C

此程式相當短，對於E類的圖形（ epicycloid ）的座標計算式為：

$$x = na\cos t - a\cos nt$$
$$y = na\sin t - a\sin nt$$

其中 n 為葉數加上 1 ，而 a 決定圖形大小， t 為 0 到 2π 的角度參數 。這種分別對 x 及 y 值以第三個變數表示的方程式，我們稱之為〃參數式〃，而第三個變數（t）即為參數 。

對於H類的圖形（ hypocycloid）計算式為

$$x = nr\cos t + a\cos nt$$
$$y = nr\sin t + a\sin nt$$

其中 n 為葉數減 1 。這兩式子牽涉到極深的數學理論而不在本書能力範圍之內 。

此程式設計的相當直接了當，主要的變數就是式子裡用到的N ，A和T（ n ，a 和 t ）。程式首先要求輸入資料，然後到對連續 t 值計算（ x ，y ）座標的程式後，並將各點依其座標畫出 。

建議練習

1. 如果在輸入葉數時試著輸入一分數，看看結果如何 。
2. 畫出如圖 6-20的圖形 。其計算式為極座標式

$$r = 2a(1 - \cos\theta)$$

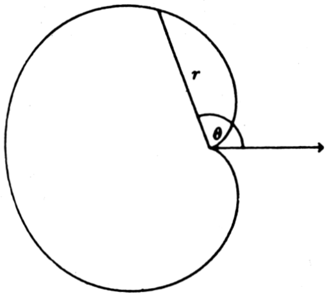


圖　6-20

3. 畫出如圖6-21的圖形。利用左圖為 $r_1 = 2a(1 - \cos\theta_1)$，右圖為 $r_2 = 2a(1 - \cos\theta_2)$ 的計算式。其中 r_1 範圍在〔$-a/2$，0〕，而 r_2 範圍在〔$+a/2$，0〕，θ_1 由正 x 軸量起，而 θ_2 由負 x 軸量起。

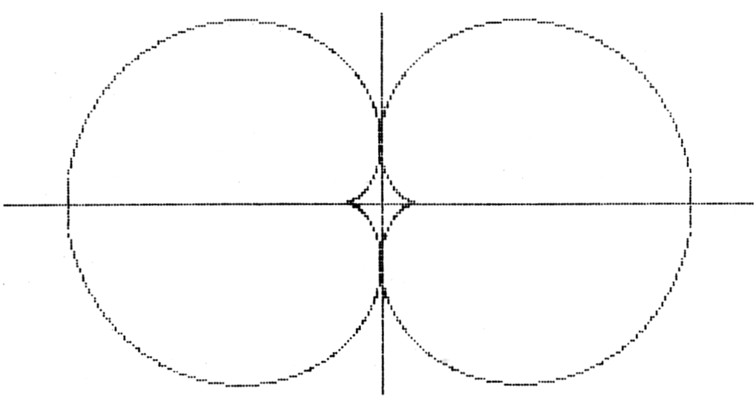


圖　6-21

4. 畫出如圖6-22的圖形。其中最外圈圖形式爲：

$$x=a(3\cos t-\cos 3t),\quad y=a(3\sin t-\sin 3t)$$

其次爲

$$x=(a/2)(3\cos t+\cos 3t), y=(a/2)(3\sin t+\sin 3t)$$

而其次爲

$$x=(a/4)(3\cos t-\cos 3t), y=(a/4)(3\sin t-\sin 3t)$$

依次類推。

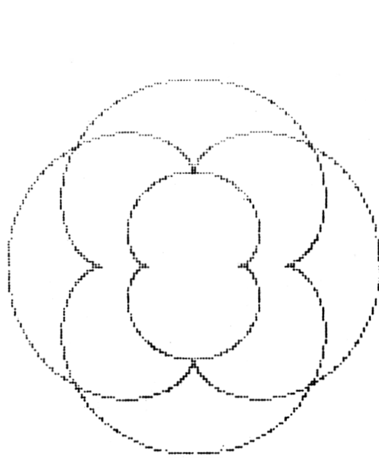


圖 6-22

ROULETTE 程式

```
100 REM *** ROULETTE-DRAWS CYCLOIDS ***
110 TEXT : HOME : HGR: HCOLOR=3
120 HOME : VTAB 21
130 PRINT "TYPE E FOR EPICYCLOID,"
140 PRINT " H FOR HYPOCYCLOID,"
150 INPUT " Q TO QUIT: ";A$
160 HOME : VTAB 21
170 IF A$="E" THEN 220
180 IF A$="H" THEN 310
190 IF A$="Q" THEN  END
200 GOTO 120
210 REM *** EPICYCLOID ***
220 INPUT "CUSPS,SIZE: ";N,A
230 N=N+1
240 IF N*A>80 THEN A=N/80
250 FOR T=0 TO 6.28 STEP .02
260 X=140+N*A*COS(T)-A*COS(N*T)
270 Y=80+N*A*SIN(T)-A*SIN(N*T)
280 HPLOT X,Y
290 NEXT T
300 GOTO 130
310 REM *** HYPOCYCLOID ***
320 INPUT "CUSPS,SIZE: ";N,A
330 N=N-1
340 FOR T=0 TO 6.28 STEP .02
350 X=140+N*A*COS(T)+A*COS(N*T)
360 Y=80+N*A*SIN(T)-A*SIN(N*T)
370 HPLOT X,Y
380 NEXT T
390 GOTO 120
```

6-8　如何畫波形

正弦波形是一種在電學中最常見的圖形（圖6-23），此種圖形一直以某固定的週期重覆下去。本節的SINEWAVE程式便可畫出此類波形。

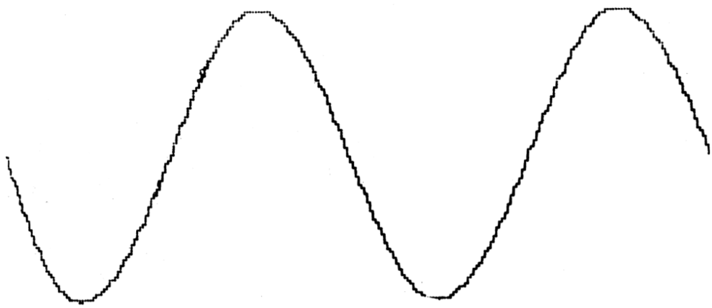


圖　6-23

SINEWAVE程式首先顯示：

```
TYPE 1 FOR SAMPLE SINE WAVES,
     2 TO SPECIFY THE PARAMETERS,
     3 TO QUIT:
```

如果你打入1，則程式在螢幕上畫出如圖6-24一樣的圖形，並在首尾兩端做標記，同時在螢幕上方顯示出此圖之方程式。在文字區內顯示出次指示表：

```
TYPE 1 FOR ANOTHER SAMPLE SINE WAVE,
     2 TO INPUT YOUR OWN PARAMTERS,
     3 TO QUIT:
```

如果你打入 1 ，則程式畫出圖6-25 ，並重新在文字區顯示出次指示表 。

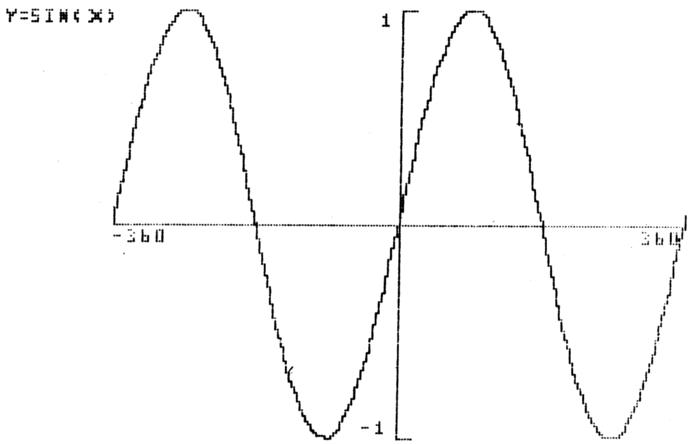


圖 6-24

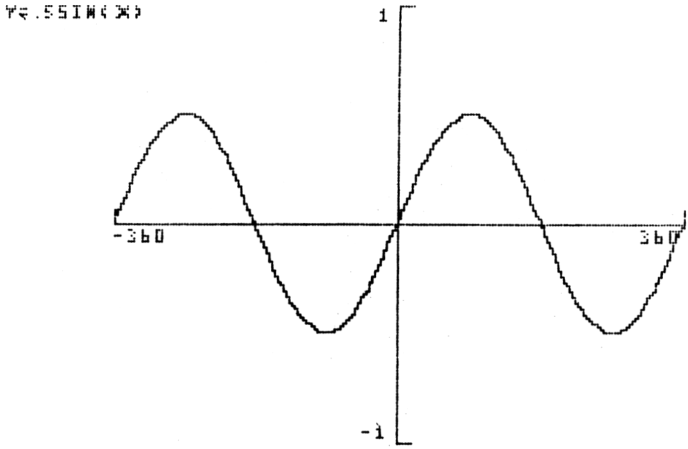


圖 6-25

如果你在圖形顯示出後繼續打入 1 ，則分別可得到圖 6-26A 及 6-26B 。

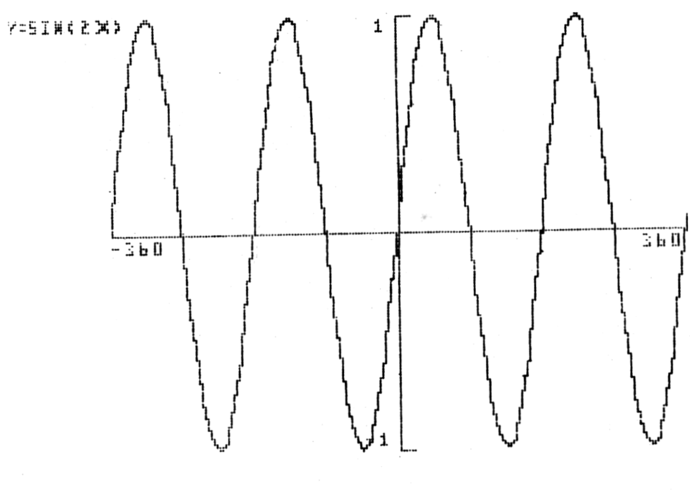


圖 6-26A

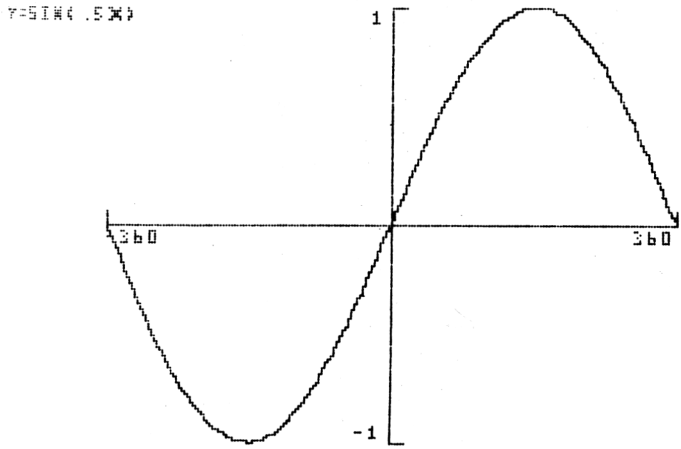


圖 6-26B

對於這些圖中，x 的範圍為 -2π 到 $+2\pi$ ，而 y 範圍為 -1 到 $+1$ 。對這二個圖形的振幅（即波峯到中間軸之距離）皆為 1 ，但圖 6-26A 在 0 到 $+2\pi$ 之間完成兩個週期，故其頻率為 2 。圖 6-26B 中在 0 到 2π 之間僅有半個週期，故其頻率為 0.5 。

要看對 x 做移動及對 y 做移動的效果，在每個圖形顯示出後繼續打入 1 ，則你可得到圖 6-27A 及 6-27B 。這兩個曲綫中 x 值仍是由 -2π 到 $+2\pi$ ，y 值仍是由 -1 到 $+1$ ，且兩者頻率都是 1 。

第一個圖形的振幅為 1 ，x 移動值為 $\pi / 2$ ，y 移動值為 0 。第二個圖形的振幅為 0.5 ，x 移動值為 0 ，y 移動值為 0.5 。所以說 x 移動則將波形水平移動而 y 移動則垂直移動波形。

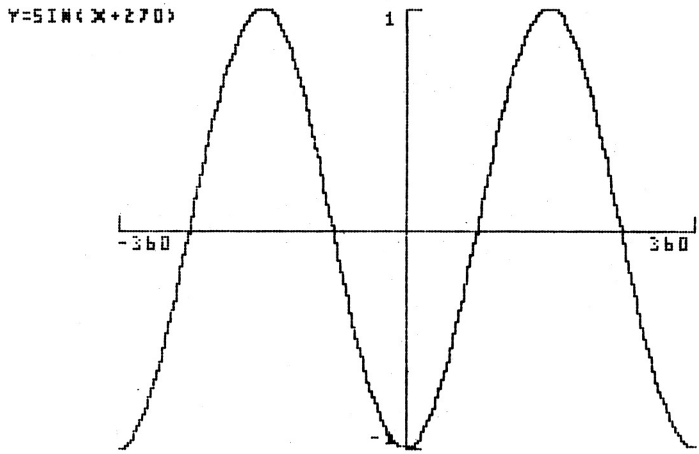


圖　6-27A

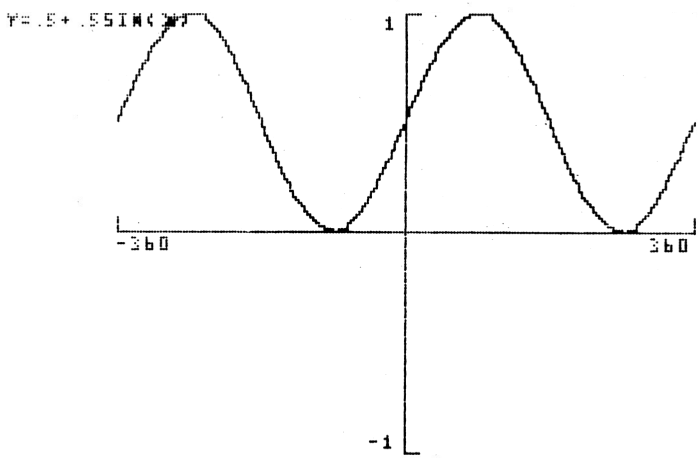


圖　6-27B

現在你可以輸入 2 來說明自己的參數。程式要求輸入振幅，頻率，*x* 位移值，*y* 位移值及 *x* 及 *y* 的取值範圍等資料。對於這些資料，程式都會先顯示其原有值並要求你輸入新值。當你完全說明了這些資料後，程式便將正弦波形畫在螢幕上，同時在上方顯示出其方程式。

程式在此同時也會顯示另一個指示表：

```
TYPE 1 TO INPUT A NEW SET OF PARAMETERS,
     2 FOR A PRESET SAMPLE,
     3 TO QUIT:
```

你可依預先設定好的參數或自行輸入參數繼續畫出各種正弦波形。

此程式簡單的說就是以螢幕中心為原點，利用 FOR 迴路的 *x* 值逐次將相對 *y* 點求出並畫在螢幕上，*y* 值由下列公式計算之：

$$y=v+a\sin(fx+h)$$

此處 *v* 代表 *y* 位移值，*a* 為振幅，*f* 為頻率，而 *h* 為 *x* 位移值。

此程式所用的主要變數為：

A	正弦波形振幅
F	其頻率
XS，YS	其 *x* 及 *y* 位移值
XV，XH	*x* 軸的範圍
YV，YH	*y* 軸的範圍
IM	預先設定好的正弦波形個數
W，H	螢幕上字元的寬及高度
XI	畫正弦波形時所用的點數

CF	將度度量轉換爲弳度量的因素
OP	主指示表的選擇數
IQ	現行畫出的預先設定正弦波數
XL，YL	字串起始點的螢幕座標
ST＄	字串
XA，YA	分別代表 y 軸的 x 座標及 x 軸的 y 座標
SX，SY	將使用者單位轉換爲螢幕座標的因素
XU，YU	正弦波上一點的使用者座標
XN，YN	上述點相對的螢幕座標
XO，YO	前一點的螢幕座標
OP	次指示表的選擇數
I	字串中字元的索引
CH＄	字串中 I 所指到的字元

此程式的運作相當直接了當。預先設定的正弦波形參數由DATA 敍述裡讀入，而由使用者定義參數時，程式會將其相對的各個原有值顯示出來並將修改的新值顯示出來，然後利用副程式畫出正弦波形。在螢幕上寫字串的副程式與 POLYPLOT 及 POLYFIT 裡的一樣，使用同樣的技巧來除去無意義的 0 ，1 及＋號，並將兩軸畫出及適當的加以標記。

建議練習

1. 容許使用者指定畫圖的間隔。
2. 容許使用者選擇是否在畫另一正弦波形時需將螢幕清除。
3. 將正弦波形以上下走向的方式在螢幕上畫出，亦即以 y 來表示 x 。
4. 畫出振幅爲20單位的正弦波形(A)依螢幕的上下緣爲軸，(B)依左右緣爲

軸(C)依螢幕的四個邊爲軸而形成一封閉曲綫，(D)依平行螢幕邊緣的各綫爲軸而形成如圖 6-28的同心圖形。

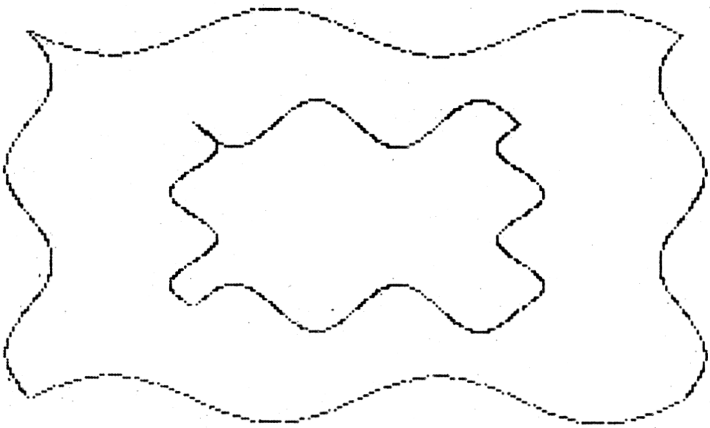


圖 6-28

5. 依圖 6-29的樣式畫出一組正弦波形。



圖 6-29

6. 依圖6-30的樣式在一組波形內以實綫填滿。

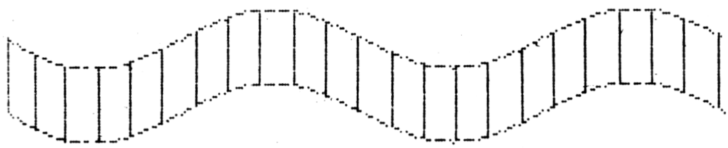


圖 6-30

7. 依圖6-31的樣式畫出一組正如不同相角的正弦波形，看起來就像扭曲的繩子。(提示：x 位値爲 π)



圖 6-31

8. 在第 7 項中圖形間以實綫填滿。(圖 6-32)

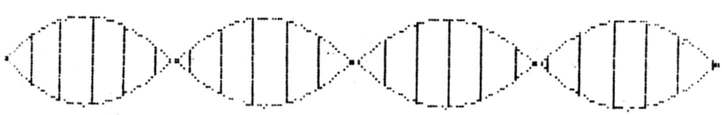


圖 6-32

9. 畫出三條不同頻率的波形，其後二條的頻率分別爲第一條的二倍及三倍，如圖 6-33 。



圖 6-33

10. 在第 9 項裡再做些參數改變來看看結果如何。

11. 先畫出兩條正弦波形而後將其 y 值相加以形成另一條曲線。

12. 畫出正弦波形中取 y 值爲其絕對值的圖形。

13. 利用12項的方法來對前面各項做實驗。

14. 畫出餘弦波形，並觀察其與正弦波形的關係。

15. 先用 SIN 函數對各值建立一正弦函數表的程式來畫出波形，比較兩者的速度。

SINEWAVE 程式　注意：如前面所述輸入時需省略 REM 敍述

```
100  REM *** SINEWAVE ***
110  LOMEM: 16384
120  IM=7: REM *** NUMBER OF PRESET SINE WAVES ***
130  REM *** AMPLITUDE, FREQUENCY, SHIFTS, AND LIMITS OF PRESET CURVES'
140  DATA 1,1,0,0,-360,360,-1;1
150  DATA .5,1,0,0,-360,360,-1,1
160  DATA 1,2,0,0,-360,360,-1,1
170  DATA 1,.5,0,0,-360,360,-1,1
```

```
180  DATA 1,1,0,270,-360,360,-1,1
190  DATA .5,1,.5,0,-360,360,-1,1
200  DATA 2,3,40,30,-200,40,36,44
210  W=5:H=4:XI=80:CF=.0174532: REM *** FONT & PLOT DATA ***
220  TEXT : HOME : HGR: HCOLOR=3
230  VTAB 21: CALL -958
240  PRINT "TYPE 1 FOR SAMPLE SINE WAVES,"
250  PRINT " 2 TO SPECIFY THE PARAMETERS,"
260  INPUT " 3 TO QUIT: ";OP
270  ON OP GOTO 290,350,1130
280  PRINT "INVALID OPTION": GOTO 230
290  IQ=0: REM *** START THE CYCLE THROUGH PRESET VALUES ***
300  VTAB 21: CALL -958
310  IF IQ>=IM THEN IQ=0: RESTORE
320  IQ=IQ+1
330  READ A,F,XS,YS,XV,XH,YV,YH
340  GOTO 470
350  VTAB 21: CALL -958: REM *** ALLOW USER TO TYPE THE PARAMETERS ***
360  PRINT "AMPLITUDE=";A;", FREQUENCY=";F
370  INPUT "NEW VALUES: ";A,F
380  VTAB 21: HTAB 11: CALL -958: PRINT A;", FREQUENCY=";F
390  PRINT "X SHIFT=";XS;", Y SHIFT=";YS
400  INPUT "NEW VALUES: ";XS,YS
410  VTAB 22: HTAB 9: CALL -958: PRINT XS;", Y SHIFT=";YS
420  PRINT "X LIMITS: ";XV;", ";XH;", Y LIMITS: ";YV;", ";YH
430  INPUT "NEW VALUES: ";XV,XH,YV,YH
440  VTAB 23: HTAB 11: CALL -958
450  PRINT XV;", ";XH;", Y LIMITS: ";YV;", ";YH
460  HOME
470  REM *** DISPLAY THE EQUATION ***
480  XL=0:YL=0
490  GOSUB 5390: REM *** DISPLAY Y ***
500  GOSUB 5370:REM  *** DISPLAY=***
510  IF YS=0 THEN 550
520  REM *** DISPLAY THE Y SHIFT ***
530  ST$=STR$(YS): GOSUB 4500
540  IF A>0 THEN  GOSUB 5350: REM *** + SIGN FOR AMPLITUDE ***
550  IF A<>-1 THEN 570
560  GOSUB 5330: GOTO 590: REM *** - SIGN FOR AMPLITUDE ***
570  IF A=1 THEN  GOTO 590: REM *** A IS 1 ***
```

```
580  ST$=STR$(A): GOSUB 4500: REM *** A IS NEITHER 1 NOR -1 ***
590  ST$="SIN(": GOSUB 4500: REM *** DISPLAY SIN( ***
600  IF F<>-1 THEN 620
610  GOSUB 5330: GOTO 640: REM *** F IS -1 ***
620  IF F=1 THEN 640
630  ST$=STR$(F): GOSUB 4500: REM *** DISPLAY F ***
640  GOSUB 5420: REM *** DISPLAY X ***
650  IF XS=0 THEN 680
660  IF XS>0 THEN  GOSUB 5350: REM *** + FOR X SHIFT ***
670  ST$=STR$(XS): GOSUB 4500: REM *** DISPLAY X SHIFT ***
680  GOSUB 5550: REM *** DISPLAY ) ***
690  REM *** PLOT THE AXES ***
700  IF XV<0 AND XH>0 THEN XA=40-200*XV/(XH-XV)
710  IF XV>0 OR XH<0 THEN XA=40
720  HPLOT XA,0 TO XA,150
730  IF YV<0 AND YH>0 THEN YA=-150*YV/(YH-YV)
740  IF YV>0 OR YH<0 THEN YA=150
750  HPLOT 40,YA TO 240,YA
760  ST$=STR$(YH): REM *** PLOT THE Y TICK MARKS AND LABELS ***
770  HPLOT XA,0 TO XA+5,0
780  XL=XA-W*LEN(ST$)-2
790  YL=1: GOSUB 4500
800  ST$=STR$(YV)
810  HPLOT XA,150 TO XA+5,150
820  XL=XA-W*LEN(ST$)-2
830  YL=148-H: GOSUB 4500
840  HPLOT 40,YA TO 40,YA-5
850  ST$=STR$(XV): REM *** PLOT THE X TICK MARKS AND LABELS ***
860  XL=40:YL=YA+2: GOSUB 4500
870  HPLOT 240,YA TO 240,YA-5
880  ST$=STR$(XH)
890  XL=240-W*LEN(ST$):YL=YA+2: GOSUB 4500
900  SX=200/(XH-XV):SY=150/(YH-YV): REM *** TO GO FROM USER TO SCREEN
     COORDS ***
910  FOR XU=XV TO XH STEP (XH-XV)/XI: REM *** PLOT THE SINE CURVE ***
920  YU=YS+A*SIN(CF*(F*XU+XS))
930  XN=40+SX*(XU-XV):YN=150-SY*(YU-YV)
940  IF XU=XV THEN 970
950  IF 40>XO OR XO>240 OR 0>YO OR YO>150 OR 40>XN OR XN>240 OR 0>YN
     OR YN>150 THE  N 970
960  HPLOT XO,YO TO XN,YN
```

```
 970  XO=XN:YO=YN
 980  NEXT XU: REM *** END OF PLOTTING CYCLE ***
1000  VTAB 21: CALL -958
1010  IF OP=2 THEN 1070
1020  PRINT "TYPE 1 FOR ANOTHER SAMPLE SINE WAVE,"
1030  PRINT " 2 TO INPUT YOUR OWN PARAMETERS,"
1040  INPUT " 3 TO QUIT: ";OQ
1050  ON OQ GOTO 310,350,1130
1060  PRINT "INVALID OPTION": GOTO 1020
1070  PRINT "TYPE 1 TO INPUT A NEW SET OF PARAMETERS"
1080  PRINT " 2 FOR A PRESET SAMPLE,"
1090  INPUT " 3 TO QUIT: ";OQ
1100  ON OQ GOTO 350,290,1130
1110  PRINT "INVALID REQUEST - PRESS A KEY"
1120  GOTO 1090
1130  HGR: TEXT : HOME : END : REM *** QUIT ***
4500  REM *** DISPLAY A STRING ***
4510  FOR I=1 TO LEN(ST$)
4530  CH$=MID$(ST$,I,1)
4540  IF CH$="0" THEN  GOSUB 5010
4550  IF CH$="1" THEN  GOSUB 5030
4560  IF CH$="2" THEN  GOSUB 5060
4570  IF CH$="3" THEN  GOSUB 5100
4580  IF CH$="4" THEN  GOSUB 5140
4590  IF CH$="5" THEN  GOSUB 5170
4600  IF CH$="6" THEN  GOSUB 5200
4610  IF CH$="7" THEN  GOSUB 5220
4620  IF CH$="8" THEN  GOSUB 5240
4630  IF CH$="9" THEN  GOSUB 5280
4640  IF CH$="." THEN  GOSUB 5310
4650  IF CH$="-" THEN  GOSUB 5330
4660  IF CH$="E" THEN  GOSUB 5440
4670  IF CH$="(" THEN  GOSUB 5530
4680  IF CH$="S" THEN  GOSUB 5470
4690  IF CH$="I" THEN  GOSUB 5490
4700  IF CH$="N" THEN  GOSUB 5510
4750  NEXT I
4760  RETURN
5000  REM *** PLOT 0 THROUGH 9 ***
5010  HPLOT XL,YL TO XL+2,YL TO XL+2,YL+4 TO XL,YL+4 TO XL,YL
```

```
5020 XL=XL+W: RETURN
5030 HPLOT XL,YL+1 TO XL+1,YL TO XL+1,YL+4
5040 HPLOT XL,YL+4 TO XL+2,YL+4
5050 XL=XL+W: RETURN
5060 HPLOT XL,YL TO XL+1,YL
5070 HPLOT XL+2,YL+1 TO XL,YL+3
5080 HPLOT XL,YL+4 TO XL+2,YL+4
5090 XL=XL+W: RETURN
5100 HPLOT XL,YL TO XL+1,YL: HPLOT XL+2,YL+1
5110 HPLOT XL+1,YL+2: HPLOT XL+2,YL+3
5120 HPLOT XL,YL+4 TO XL+2,YL+4
5130 XL=XL+W: RETURN
5140 HPLOT XL,YL TO XL,YL+2 TO XL+2,YL+2
5150 HPLOT XL+2,YL TO XL+2,YL+4
5160 XL=XL+W: RETURN
5170 HPLOT XL+2,YL TO XL,YL TO XL,YL+2 TO XL+1,YL+2
5180 HPLOT XL,YL+4 TO XL+1,YL+4: HPLOT XL+2,YL+3
5190 XL=XL+W: RETURN
5200 HPLOT XL,YL TO XL,YL+4 TO XL+2,YL+4 TO XL+2,YL+2 TO XL,YL+2
5210 XL=XL+W: RETURN
5220 HPLOT XL,YL TO XL+2,YL TO XL,YL+4
5230 XL=XL+W: RETURN
5240 HPLOT XL+1,YL TO XL+2,YL TO XL+2,YL+1
5250 HPLOT XL,YL+1: HPLOT XL+1,YL+2: HPLOT XL,YL+4
5260 HPLOT XL+1,YL+4 TO XL+2,YL+4 TO XL+2,YL+3
5270 XL=XL+W: RETURN
5280 HPLOT XL,YL+4 TO XL+2,YL+4 TO XL+2,YL TO XL,YL TO XL,YL+2
     TO XL+2,YL+2
5290 XL=XL+W: RETURN
5300 REM *** PLOT  -+=YXE ***
5310 HPLOT XL+2,YL+4
5320 XL=XL+W: RETURN
5330 HPLOT XL,YL+2 TO XL+2,YL+2
5340 XL=XL+W: RETURN
5350 HPLOT XL,YL+2 TO XL+2,YL+2: HPLOT XL+1,YL+1 TO XL+1,YL+3
5360 XL=XL+W: RETURN
5370 HPLOT XL+1,YL+1 TO XL+3,YL+1: HPLOT XL+1,YL+3 TO XL+3,YL+3
5380 XL=XL+W: RETURN
5390 HPLOT XL,YL TO XL+2,YL+2 TO XL+4,YL
5400 HPLOT XL+2,YL+4 TO XL+2,YL+2
```

```
5410  XL=XL+W: RETURN
5420  HPLOT XL,YL+4 TO XL+4,YL: HPLOT XL,YL TO XL+4,YL+4
5430  XL=XL+W+2: RETURN
5440  HPLOT XL+2,YL TO XL,YL TO XL,YL+4 TO XL+2,YL+4: HPLOT XL,YL+2
      TO XL+2,YL+2
5450  XL=XL+W+1: RETURN
5460  REM *** PLOT SIN() ***
5470  HPLOT XL+2,YL TO XL,YL TO XL,YL+2 TO XL+2,YL+2 TO XL+2,YL+4
      TO XL,YL+4
5480  XL=XL+W: RETURN
5490  HPLOT XL,YL TO XL+2,YL:HPLOT XL,YL+4 TO XL+2,YL+4:HPLOT XL+1,YL
      TO XL+1,YL+4
5500  XL=XL+W: RETURN
5510  HPLOT XL+2,YL TO XL+2,YL+4 TO XL,YL TO XL,YL+4
5520  XL=XL+W: RETURN
5530  HPLOT XL+1,YL+4 TO XL,YL+2 TO XL+1,YL
5540  XL=XL+W: RETURN
5550  HPLOT XL,YL+4 TO XL+1,YL+2 TO XL,YL
5560  XL=XL+W: RETURN
```

第七章　更深入的應用

電腦繪圖的領域裡有許多發展的方向，此章將介紹三項主題，而你可依自己的興趣深入研究之。其一爲物體的三維投射圖，其二爲在動態或靜態圖片中如何使用色彩，及使電腦繪圖容易設計及使程式執行速度更快的數學知識。

7－1　物理投射圖

我們現在來用 ZVIEW 程式來將物體的立體圖形畫成如圖 7-1 所示。此程式首先要求輸入描述立方體的資料，或任何其他由平面和直線構成的實體資料。 ZVIEW 將螢幕中點視爲一三維空間 xyz 座標系的原點，水平爲 x 軸，垂直爲 y 軸，z 軸的方向爲遠離你的直線方向。（參見圖 7-

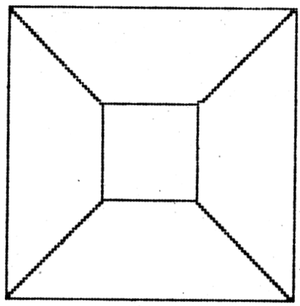


圖 7-1

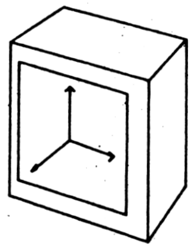


圖 7-2

此程式假設你正視通過螢幕中心而與螢幕距離爲此螢幕高之五倍，如圖 7-3 所示。

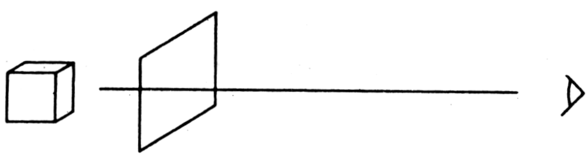


圖 7-3

我們由 ZVIEW 來定義一實體：

1. 物體角依任何順序的座標。程式分別依輸入順序將各角編號爲 1，2，3，…，除非你自己要定義這些數。
2. 程式用兩角的編號來代表一邊。

此程式先顯示出主指示表：

```
TYPE:
1 TO DEFINE DISTANCES, 2 TO DEFINE POINT,
3 TO DISPLAY EDGE, 4 TO CLEAR SCREEN,
5 TO RESTART, 6 TO QUIT:
PLEASE TYPE OPTION:
```

如果你打入 1 ，則程式顯示出次指示表：

```
TYPE 1 FOR POSITION OF EYE,
     2 FOR DISTANCE TO SCREEN,
     3 FOR HEIGHT OF SCREEN,
     IN GLOBAL DIMENSIONS:
```

在你適當的輸入選擇號數後，程式會顯示出你要求的現行大小值。你輸入新值來取代它，然後又回到上指示表。

要說明某一角的座標，在上指示表內打入 2 。程式要求你輸入此點的編號，你輸入後，程式繼續要求其座標。要定義一邊，在上指示表裡打入 3 ，程式便要求輸入此邊的兩端點之編號。在你輸入此資料後，程式便將其座標顯示出來以讓你選擇是否要畫出此邊。選擇功能 4 讓你在不洗掉已存入的點座標的情況下將螢幕清除。選擇功能 5 及 6 分別使你能重新開始整個程序或離開此程式。

本程式用的幾何理論相當簡單（參見圖 7-4 ）。此圖說明一長 L 的綫段，垂直 z 軸並與 x 軸平行。其在螢幕上的投射圖（由眼睛看去）便依比例縮小，如此長爲 l ，則由相似三角形理論得知：

$$\frac{l}{D}=\frac{L}{(R+D)} \qquad \text{hence} \qquad l=\frac{L}{(1+R/D)}$$

此處 R 及 D 分別爲眼睛到綫段和到螢幕的距離。現在如果我們考慮圖 7-5 中的一綫段垂直 z 軸而與 y 軸平行，高度爲 h，則其投影量 h 爲

$$h=\frac{H}{(1+R/D)}$$

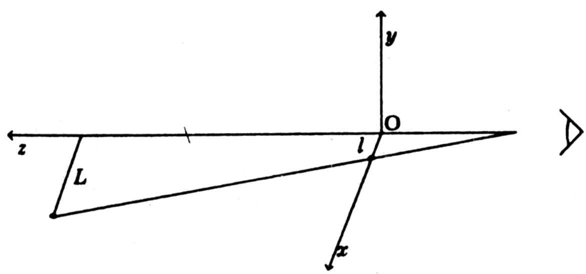


圖 7-4

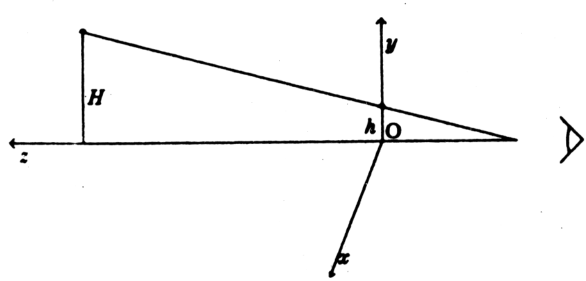


圖 7-5

依此原理得知如果一點（x，y，z）投影到螢幕上時，其座標變爲

$$x_s=\frac{x}{(1+z/D)},\qquad y_s=\frac{y}{(1+z/D)}$$

這便是程式中將 z 軸轉換成二維投射圖的基本方程式。程式利用一個如此的 FOR 迴路來將各個三維點座標轉換成二維的螢幕座標。另一 FOR 迴路則處理由兩點連成的各綫，並依所需將其畫出。

本程式用到的主要變數爲：

XG（50）	某點的 x 座標
YG（50）	某點的 y 座標
ZG（50）	某點的 z 座標
IM%	程式所能處理的最大點數
ID%	某點的編號
ZE	眼睛到原點距離
D	眼睛到螢幕距離
S	螢幕高度
F	將轉換過的 x，y 座標再轉換爲螢幕座標的因素
I1%	正在畫的某一邊的一端點編號
I2%	上述一邊另一端點編號

建議練習

1. 讓使用者輸入位在 yz 平面上的數個三角形的資料而將此立方體畫出。
2. 相對第 1 項來處理金字塔的圖形。

ZVIEW 程式

```
100 REM *** ZVIEW FOR PERSPECTIVE PROJECTION ALONG Z AXIS ***
110 DIM XG(50),YG(50),ZG(50)
120 IM%=50:ID%=1
130 ZE=20:D=5:S=15
140 HOME : HGR : HCOLOR=3: POKE 34,20
150 HOME : PRINT "TYPE: "
160 PRINT "1 TO DEFINE DISTANCES, 2 TO DEFINE POINT";
170 PRINT "3 TO DISPLAY EDGE, 4 TO CLEAR SCREEN,"
180 INPUT "5 TO RESTART, 6 TO QUIT: ";OP
190 HOME : ON OP GOTO 220,350,440,140,120,610
200 PRINT : FLASH : PRINT "INVALID OPTION: ";OP;", PRESS ANY KEY"
210 NORMAL : GET A$: GOTO 150
220 HOME : PRINT "TYPE 1 FOR POSITION OF EYE,"
230 PRINT "      2 FOR DISTANCE TO SCREEN,"
240 PRINT "      3 FOR HEIGHT OF SCREEN,"
250 INPUT "      IN GLOBAL DIMENSIONS: ";OQ
260 HOME : ON OQ GOTO 290,310,330
270 PRINT : FLASH : PRINT "INVALID REQUEST: ";OQ;", PRESS ANY KEY"
280 NORMAL : GET A$: GOTO 220
290 PRINT "DISTANCE FROM EYE TO ORIGIN (NOW=";ZE;"): ";
300 INPUT "";ZE: GOTO 150
310 PRINT "DISTANCE FROM EYE TO SCREEN (NOW=";D;"): ";
320 INPUT "";D: GOTO 150
330 PRINT "HEIGHT OF SCREEN (NOW=";S;"): ";
340 INPUT "";S: GOTO 150
350 PRINT "ID # OF POINT(NEXT FREE #=";ID%;"): ";
360 INPUT "";JD%
370 IF JD%>0 AND JD%<IM% THEN 410
380 PRINT : FLASH : HTAB 10: PRINT "INVALID ID#": NORMAL
390 PRINT "PRESS ANY KEY TO RETURN TO MAIN MENU"
400 GET A$: GOTO 150
410 INPUT "COORDINATES: ";XG(JD%),YG(JD%),ZG(JD%)
420 IF JD%=ID% THEN ID%=ID%+1
430 GOTO 150
```

```
440  INPUT "ID #S OF END POINTS: ";I1%,I2%
450  IF 0>I1% OR I1%>IM% OR 0>I2% OR I2%>IM% THEN 380
460  HOME
470  PRINT "COORDINATES: ";XG(I1%); SPC( 3);YG(I1%); SPC( 3);ZG(I1%)
480  PRINT  SPC( 13);XG(I2%); SPC( 3);YG(I2%); SPC( 3);ZG(I2%)
490  INPUT "TYPE 1 TO DISPLAY, 2 TO SKIP: ";OQ
500  IF OQ<>1 THEN 150
510  F=(150*D)/(S*(ZE-ZG(I1%)))
520  X1=140+F*XG(I1%):Y1=75-F*YG(I1%)
530  IF 0<=X1 AND X1<=279 AND 0<=Y1 AND Y1<=159 THEN 550
540  PRINT "FIRST POINT OUT OF RANGE: ";X1,Y1: GOTO 390
550  F=(150*D)/(S*(ZE-ZG(I2%)))
560  X2=140+F*XG(I2%):Y2=75-F*YG(I2%)
570  IF 0<=X2 AND X2<=279 AND 0<=Y2 AND Y2<=159 THEN 590
580  PRINT "SECOND POINT OUT OF RANGE: ";X2,Y2: GOTO 390
590  HPLOT X1,Y1 TO X2,Y2
600  GOTO 150
610  TEXT : HOME : END
```

7—2　TRYTOR程式

本節的TRYTOR程式畫出一組如圖 7-6A 及 7-6B 的圖形。其中小圓碟的半徑爲 a ，而所有圓碟的圓心都在半徑爲 b 的圓周上。此程式由四個參數 a ， b ， θ 及 ϕ 分別來代表小圓半徑，大圓半徑，極角和視者的方位角。對於不同的值，產生不同的立體圖效果（如圖 7-6A 及 7-6B ）。同時此程式假設視者距離螢幕很遠！

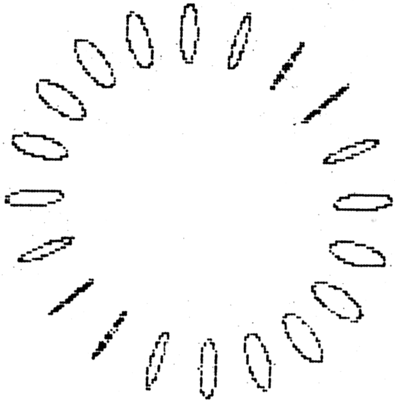


圖 7-6A　$a=15,\ b=40,\ \theta=30,\ \phi=30.$

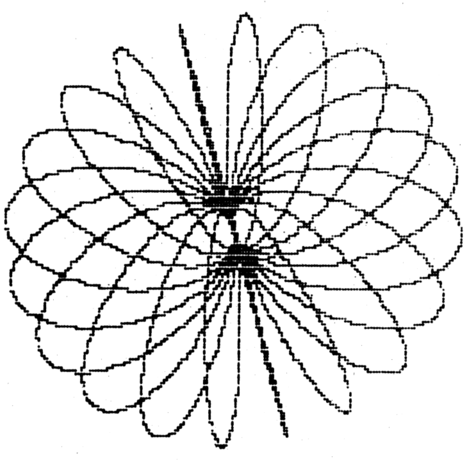


圖 7-6B　$a=40,\ b=40,\ \theta=30,\ \phi=45.$

此程式首先要求輸入(1) a ，b ，θ 及 ϕ 值(2)畫出圖形的投射圖及(3)再要求使用者是否繼續畫另一組圖。

程式裡用到的主要變數爲：

X (I)	每一小圓碟上某點的 x 座標
Y (I)	上述點之 y 座標
A	圓碟半徑
B	大圓半徑
TH , PH	分別爲 θ 及 ϕ
J	在大圓一周上取 1 到 100 的值
Z1 , … , Z4 , CP , SP , U1 , … , U4	中間值

此程式用 a ，b 輸入値來對圓碟建立 X (I)及 Y (I)列陣。系統中使用的方程式爲：

$$x = X(I)\cos(J/2\pi), \qquad y = Y(I), \qquad z = X(I)\sin(J/2\pi)$$

同一點的投射點（ x_r ，y_r ）爲

$$x_r = x\cos\theta\cos\phi + y\cos\theta\sin\phi - z\sin\theta$$
$$y_r = -x\sin\phi + y\cos\phi$$
$$z_r = x\sin\theta\cos\phi + y\sin\theta\sin\phi + z\cos\theta$$

因爲我們只需對 z 軸的投射圖，所以（ x_r ，y_r ）座標便足夠了。程式對每個圓碟共畫出 100 點，而大圓內含20個小圓碟。

建議練習

1. 容許使用者說明大圓中小圓碟數目及用來描繪此圓碟的點數。
2. 使數個類似圖形同時出現在螢幕上。
3. 使程式系統的執行一連串 θ 及 ϕ 值以觀察結果。
4. 在程式開頭用別的方法加速 sin 及 cos 的運算。

TRYTOR 程式

```
100  REM *** TRYTOR PROGRAM TO DRAW TOROID ***
110  DIM CS(101),SN(101),X(101),Y(101)
120  REM *** SET UP TABLE OF SINES&COSINES ***
130  DN= 0628:AN=0
140  FOR I=1 TO 100
150  CS(I)=COS(AN):SN(I)=SIN(AN):AN=AN+DN
160  NEXT I
170  HGR : HCOLOR=3: HOME : VTAB 21
180  INPUT "ENTER A,B,THETA,PHI: ";A,B,TH,PH
190  REM *** CREATE PROTOTYPE CIRCLE ***
200  FOR I=1 TO 100
210  X(I)=B+A*CS(I):Y(I)=A*SN(I)
220  NEXT I
230  REM *** FORMSIN&COSOF THETA&PHI ***
240  Z1=COS(TH):Z2=SIN(TH):Z3=COS(PH):Z4=SIN(PH)
250  REM *** FORM SUCCESSIVE RINGS ***
260  FOR J=1 TO 100 STEP 5
270  CP=CS(J):SP=SN(J)
280  U1=Z1*Z3*CP-Z2*SP:U2=Z1*Z4
290  U3=-Z4*CP:U4=Z3
295  HPLOT 140+U1*X(1)+U2*Y(1),U3*X(1)+U4*Y(1)+80
300  FOR K=1 TO 100
310  HPLOT  TO U1*X(K)+U2*Y(K)+140,U3*X(K)+U4*Y(K)+80
320  NEXT K
330  NEXT J
```

```
340  PRINT "TYPE E TO END";: GET A$: IF A$="E" THEN  END
350  PRINT
360  INPUT "CLEAR SCREEN?(Y OR N) ";A$: IF LEFT$(A$,1)="N" THEN 180
370  GOTO 170
```

7–3 萬花筒

彩色圖形一直是電腦繪圖裡重要而有趣的主題。本書裡所有的程式都可加入顏色而使程式及圖形更有價值。KALEID程式便能在彩色螢幕上造成如萬花筒的圖形，即使是在黑白顯示器上，同樣能產生一直改變的有趣圖案。圖 7-7 裡用一些符號來代表各種不同的顏色（因為本書非彩色印刷，此為低解析度的繪圖，40 × 40 格的圖形以對稱而快速的方式一直保持變動。程式用兩個重疊 FOR 迴路來設定各點的 COLOR 值，同時計算出的方格座標也需取其整數。SCRN函數功能是找出某一點的顏色值，此迴路索引分別為 C 和 D，也是程式裡僅有的變數。

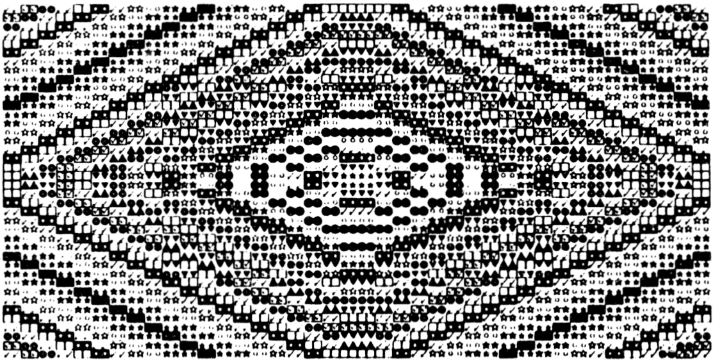


圖 7-7

建議練習

1. 應用上述原理到不同的圖形上。
2. 依不同的對稱方式創造出不同的效果。

KALEID 程式

```
100 REM *** KALEID-USES LO-RES GRPAHICS TO CREATE COLORFUL PATTERNS ***
110 GR
120 REM *** PROGRESS INWARD TO CENTER  OF SCREEN ***
130 FOR C=0 TO 19
140 FOR D=C TO 20
150 COLOR=ABS(15*SIN(C+D)): GOSUB 340
160 NEXT D,C
170 REM *** PROGRESS TO EDGES OF SCREEN ***
180 FOR C=19 TO 0 STEP -1
190 FOR D=C TO 20
200 COLOR=ABS(15*COS(C+D)): GOSUB 340
210 NEXT D,C
```

```
220  REM *** GO BACK IN ***
230  FOR C=0 TO 19
240  FOR D=C TO 20
250  COLOR=15- SCRN( C,D): GOSUB 340
260  NEXT D,C
270  REM *** COME BACK OUT ***
280  FOR C=19 TO 0 STEP -1
290  FOR D=C TO 20
300  COLOR= SCRN( C,D)+1: GOSUB 340
310  NEXT D,C
320  GOTO 230: REM *** CONTINUE CYCLE ***
330  END
340  REM *** DRAW 8 POINTS, WITH SYMMETRY AROUND THE SCREEN'S CENTER ***
350  PLOT C,D: PLOT D,C: PLOT 39-C,D: PLOT 39-D,C
360  PLOT 39-C,39-D: PLOT 39-D,39-C: PLOT C,39-D: PLOT D,39-C
370  RETURN
```

7—4　一些簡益幾何學

本節討論的簡易幾何知識可能你在學校裡都學過，這裡祇是爲充實電腦畫圖的基礎而將其列出以供參考。

直線方程式：

$$y=mx+b$$

此處m爲斜率而b爲此線在y軸的截距（圖7-8 ）。可能你已用此方程式設計過程式，但需注意的是對於垂直線，此式並不正確，所以必須將垂直線視爲一種特例。另一種定義直線的方程式爲：

$$Ax+By+C=0$$

其中A及B爲此綫與x軸正向與原點到此綫的法向量間的夾角。而C爲法綫長（參見圖 7-9）。

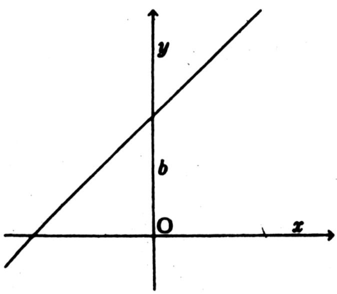


圖 7-8

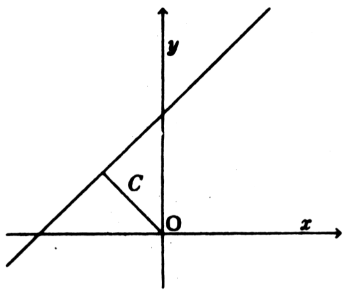


圖 7-9

第八章　Applesoft 語言簡述

8－1　APPLE 裡的語言

你可在 Apple II 用許多不同的語言設計程式。其中 Applesoft(即本書程式所用的語言) 及 Integer BASIC 爲 Dartmouth 大學的原始 BASIC 發展出的兩種 BASIC 語言。在 Apple II 裡也可用 6502 組合語言及附加上一些語言卡後的 PASCAL 和 FORTRAN 等語言來寫程式。同時 Apple 上也可用如 DOS 及 CP／M 等強而有力的作業系統來做各種不同的應用程式。本書裡程式理論的一些相關資料可由下列各參考書籍中找到：

(1)　The Applesoft Tutorial

(2)　Apple II 參考手册

(3)　BASIC 程式設計參考手册

(4)　磁碟作業系統手册 (DOS Manual)

8－2 APPLESOFT程式

Applesoft 是一種由原始 BASIC 加以擴充的一種功能強大的語言。其程式可直接一行行（每行至多255 字元）輸入後而直接執行之。每一程式敍述行必須有一個在 0 到 65536 之間的行數。同一行裡可用分號將數個敍述分開。輸入一行後按RETURN鍵便結束此行。所有用來解釋程式作用的文字說明可放在 REM 敍述裡。

變數及常數種類有實數，整數及字串。一個變數可用一字元表示（如 X），或可用一字元後接數數字或一字元（例如 AB ， G7 ）表示。變數各稱後接％符號代表此變數爲整數（例如Q％，PG％，H5％），後接＄符號代表此爲一字串（例如D＄，EZ＄，F2＄）。字串常數以引號加上字串來表示（例如" EUREKA "）。字串長度不定，至多可達 256 個字元。一維或多維列陣需用 DIM 敍述事先宣告，例如：

```
DIM A(5),E%(10,7),G%(3,5),B(2,4,3)
```

列陣變數的註標以 0 做開始；亦即此敍述你使用A（0）到A（5）及E%（0，0)到E%（10，7）。最大維數爲88，同時同一名稱可分別代表列陣變數及非註標變數（例如A，E%，G%及 B 可同時用在含上面 DIM 敍述的程式裡）。整數列陣所佔的記憶空間較實數列陣小。超過兩個字元的名稱雖不會產生錯誤，但系統並不對其加以區分（例如SCALE，LIMIT%及TITLE%與SC，LI%及TI%對系統而言是同一名稱）。在迴路裡不用常數而用變數將增快執行速度。

8－3 初設定及指定

程式裡所有的數值及字串變數在程式開始執行時都分別初設定爲 0 和空字串。CLEAR 敍述重覆此初設定功能。變數取值的方法有：

1. 用＝號的指定敍述（可含 LET ）。（例如 $x = 5$ ；LET $y = 5$ ）
2. 輸入敍述（ 8-6 節）
3. DATA 敍述，其形式爲

DATA以逗點分開的數值列表

DATA 敍述裡的各值可用READ 敍述指定到各變數裡，其形式爲：

READ 以逗點分開的變數列表

例如若程式開頭爲：

```
100 DATA 1,2,3,4,5
110 READ A,B,C,D,E
```

則將變數 A 到 E 分別指定數值爲 1 到 5 。程式裡經常會用一個READ 敍述來讀入在 DATA 敍述裡的資料，但惟一需注意的是各個變數與其資料間的順序需保持著，以避免讀錯資料，例如：

```
100 READ A,B: DATA 1,2,3: READ C
110 DATA 4
120 DATA 5: READ D,E
```

與前面的例子具相同的效用，同時這個讀入資料的功能並不因第 100 及110 行之間，或第 120 行之間有其他程式敍述與否而改變，祇要這些敍述裡不再含READ 或 DATA 敍述即可。

敍述

```
RESTORE
```

則將READ 的變數重新取DATA敍述的最前一個資料。

8—4 數值及文字陳式 (EXPRESSIONS)

在算術陳式中，符號＋，－，＊，／和 (CHIFT - N鍵) 分別代表加，減，乘，除和指數運算。

如果 x 代表一數值陳式 (即此陳式之結果爲一數值而非字串) 時，函數 INT (x)值爲取出 x 的整數部份 (例如 x 爲 1.5，3，－2.5，－4，則INT (x)分別爲 1，3，－3，－4 ，請注意 INT (x)對 x 爲負數時之處理)。函數 SGN (x)視 x 值爲正，零或負分別取爲＋1，0 或－1 。ABS (x) 及SQR (x)分別取 x 的絕對值和平方根値。函數 RND (x) 則在 0 到 0.999 … 9 之間取一隨機數，如果 x 爲正數，則 RND(x)與前一 RND 函數値必不相同。如果 x 爲零，則與前一 RND 値相同，如果 x 爲負時，RND(x) 値由 x 値惟一決定。函數 SIN，COS及 TAN 分別將 x (即其函數變數) 以弳度量計算其正弦，餘弦及正切値。函數 ATN(x)則取 x 之反正切値 ($\tan^{-1}(x)$)，其範圍在 $-\pi/2$ 到 $+\pi/2$ 之間，單位爲弳。欲將弳度量角轉換成度度量角，只需乘以 57.2958 即可。函數 EXP(x)及 LOG (x) 分別取 x 的自然指數及自然對數值。

函數 STR$ (x) 將一實數或整數轉換成一字串。如果 a $ 代表一字串

陳式，VAL(*a*$) 取 *a*$ 所代表的實數或整數值（例如，VAL("123GO")等於整數值 123 ）。函數 ASC(*a*$)取 *a*$ 中第一個字元的 ASCII 碼。函數 CHR$(*n*)，當 *n* 爲 0 到 255 之間的一整數時），取 *n* 所代表的ASCII字元。如果 *a*$，*i*，*j*分別代表一字串陳式及兩數值陳式，則 LEN(*a*$)，LEFT$(*a*$，*i*)，RIGHT$(*a*$，*i*)，MID$(*a*$，*i*)及 MID$(*a*$，*i*，*j*)分別取出 *a*$ 的長度，*a*$ 左邊 *i* 個字元，*a*$ 右邊 *i* 個字元，*a*$ 字串以第 *i* 個字元起直到右端的子字串及 *a*$ 中第 *i* 個字元起的 *j* 個字元。

除了上述內建的函數之外，Applesoft 容許使用者自行定義函數，其方式爲：

DEF FN 函數名稱(假變數)=陳式

其中右邊的陳式必須含假變數，同時系統只以函數名稱中的頭兩個字元區分之，正如系統用來區分變數名稱一樣。例如我們如果要定義X mod 10（即函數將 *x* 除以10後取其餘數）的函數可定義如下：

```
DEF FN MD(X)=INT((X/10-INT(X/10))*10+.05)*SGN(X/10)
```

下列敍述

```
A=FN MD(17): B=FN MD(-25)
```

則設定變數A爲 7 ，B爲 −5 。MD 函數也可修改成X mod Y，你祇需將定義裡的10以 *y* 取代即可。

PEEK(*m*)函數取出記憶位置*m*的內容，此指令用途極爲廣泛，可用來檢視系統使用的特殊位置的內容以了解整個 Apple 的情況。指令 POKE *m*，*n* 則將記憶位置*m*的內容設定爲 *n*，其中*m*，*n* 均爲整數陳式。

8－5 流程控制

下列的各敍述具有改變程式流程的功能：

1. GO TO 行數

（GO TO 也可打成GOTO）

2. ON 整數陳列GOTO 行數列

（如果此整數陳式值不在各行數列之間，則程式錯誤）

3. GOSUB 行數

（此敍述跳到以行數爲首的副程式，注意在跳到副程式之前必須先做好各參數的設定工作，並在跳回主程式後適當取出結果值）

4. ON 整數陳式 GOSUB 行數列

5. RETURN

6. POP

（此敍述使下一個RETURN敍述跳到次層回復地址）

7. FOR 索引＝上限 TO 下限［STEP 增量］

（此爲一般語言中 DO 迴路的 Applesoft 形式，各上下限及增量可以陳式表示之）

8. NEXT

（此結束 FOR 迴路）

9. CALL 系統記憶體絕對地址

（此節中將提到一些例子）

10. ONERR GO TO 行數

（發生錯誤時跳到行數指定的敍述，發生錯誤的行數由下列式子決

定。

```
PEEK(218)+PEEK(219)*256
```

此錯誤的種類以數值碼表示則爲

```
PEEK(222)
```

其中包括 5（輸入檔結束），42（READ 指令讀不到資料），53（不合法量），69（溢量），77（不正確的註標），133（除零），176（字串太長）及 254（不正確的輸入），詳細的錯誤碼及其含義可參見參考書籍(3)）

11. RESUME

（在 ONERR 敍述轉移後，此敍述將控制交與產生錯誤的下一行）

12. IF 條件 THEN 敍述

如果此敍述爲 GOTO 敍述，則可省略 GOTO 而只打出行數。如果條件是個數值陳式，則結果不爲 0 時代表條件爲眞，反之爲 0 時代表假。條件也可是其他由各種關係符號組成的邏輯陳式（即只有眞與假爲其結果），其中符號＜，＜＝，＝，＞＝，＞ 都很明顯，而符號＜＞則代表不等於。另外邏輯陳式也可與邏輯運算 AND，OR 及 NOT 等相結合。當此條件成立時程式便執行 THEN 後面的敍述，否則繼續下一行敍述。例如 x 爲 5，

```
IF X=5 THEN A=2: B=3: PRINT C
```

將 A 設定爲 2，B 爲 3，並印出 C。這種敍述在設計程式時非常有用，但也需小心。

8–6 I O指令

敍述如

PRINT data-list

將資料輸出到顯示器上，除非(1)你預先用PR＃ 敍述（此將稍後做解釋）指定到其他裝置上，或(2)如8-8 節所述此輸出是對磁碟機而言。PRINT 的格式除非前一個 PRINT 敍述尾端用了一個分號，否則都將以新的一行做開始，印出的資料可爲任何數値，變數或字串陳式（包括空字串），在各項資料間用一分號則可使輸出不做間隔，在 PRINT 敍述尾端用一分號使下一個 PRINT 敍述繼續在同一行接著做輸出。資料項間用逗號（，）則使各資料印出位置依序爲第 1 ，16及32第螢幕格。

資料列裡也可使用 SPC（ *i* ） 及 TAB（ *i* ） 函數來設計輸出格式，此二函數的功能分別爲預留 *i* 個空格及將光點 TAB 到第 *i* 格。

指令

HTAB *i*　　VTAB *i*　　HOME

分別將光點移到現行行的第 *i* 格，將光點移到第 *i* 行的第 1 格及將光點移到第 1 行的第 1 格。HOME 指令同時會清除螢幕。在HTAB 及VTAB 指令中，*i* 的取值必須在螢幕範圍之內。

函數

POS（假陳式）

將取回光點的水平位置，假陳式只是用來將括號分開的任何一數。光點的

水平位置也可用 PEEK (36)取得，而光點的垂直則可用PEEK(37)取得。注意這些值的計算方法和HTAB 或 VTAB 不同， POS 或PEEK 取得的數值是從 0 開始算起。

指令 INVERSE，FLASH 及 NORMAL分別將此後顯出的字元以〝相反模式〞（即白底黑字），〝閃爍〞（即黑白交互顯示）及〝正常模式〞（即黑底白字）的形式顯示在螢幕上。

指令

SPEED＝整數陳式（值在 0 到 255 之間）

控制輸出字元的速度。

敘述

PR ＊ *n*

將輸出送到*n*所代表的輸出裝置（ *n* 取值爲 1 到 7 ），如印表機。而 PR ＊ 0 則回到螢幕上。

敘述

INPUT〔〝prompt〞〕；資料列

在螢幕的下一行開始處印出prompt（可爲任何通知使用者的字串，如果省略的話則印出一問號），然後等待使用者做輸入，並將各輸入值依資料順序指定給各個變數。輸入的方法通常由鍵盤打入，或是由類似 PR ＊ 的 IN ＊ 敘述指定的裝置輸入，或是由磁片中讀入。資料列中同樣的以逗點分開各資料項。

敘述

STORE 列陣名稱

可用來將此列陣存到連在Apple II的錄音帶中。如欲重新取出此資料則用敍述

RECALL 列陣名稱

指令

GET *a*$

常用在程式提供一段暫停以便使用者能觀看螢幕上的結果的情形裡。其中 *a*$ 代表一字串變數。當使用者按下任一鍵後，*a*$ 便被設定爲此字元（但螢幕上並不印出），然後程式繼續執行。如果在 Apple 還未執行到此 GET*a*$ 指令時你已按下一鍵，則 *a*$ 同樣取此字元，但程式却不暫停。WAIT 指令也可用來達到這種目的（參見參考書籍(3)）。

```
POKE 32,L   POKE 33,W   POKE 34,T   POKE 35,B
```

是分別用來設定文字區的左邊，寬度，上邊及底邊的指令。其中L，W，T及B分別代表螢幕的格數。

CALL -958 and CALL -868

可用來將光點後的同一行內所有文字清除或將光點後直到螢幕底部的所有文字清除。你也可用下列兩敍述來達到同樣的效果。

PRINT CHR$(27);"F" and PRINT CHR$(27);"E"

指令 PDL（1）及 PDL（2）分別用來輸入搖桿值。

```
X=PEEK(-16336)
```

是用來控制喇叭的頻率。

8-7 繪圖指令

指令

```
GR
```

將 Apple 轉換到低解析度繪圖模式，此模式之螢幕解析度爲 40 × 40 格。x，y 座標分別爲水平由左到右和垂直由上到下的 0 到 39。

指令

```
PLOT x,y
HLIN x1,x2 AT y
VLIN y1,y2 AT x
```

分別在（x，y）上畫一點，由 x_1 畫條水平綫到 x_2（其高度爲 y）和由 y_1 畫一條垂直綫到 y_2（距離螢幕左邊 x 處）。

指令

```
COLOR=n
```

設定繪圖顏色，而 n 在 0 到15之間。系統一般設定顏色爲 0（黑色）。各數值所代表的顏色如下：

0	黑	8	棕
1	紫紅	9	橙
2	深藍	10	灰
3	紫	11	粉紅

4	深綠	12	綠
5	灰	13	黃
6	藍	14	碧綠
7	淡藍	15	白

在黑白顯示器上，不同的顏色則不同的深暗度來表示。

在高及低解析度繪圖模式裡，系統通常在螢幕底部預留四行文字區。在低解析度裡可用POKE 指令

```
POKE -16302,0
```

將此文字區除去而使 y 值擴充到 47 。

上述的相反效果則由下列指令達到。

```
POKE -16301,0
```

指令

```
SCRN(x,y)
```

取回（ x ， y ）點的顏色值，其中參數可爲任何合法的數值陳式。

指令

```
HGR
```

將 Apple 轉換到高解析度繪圖模式，此時螢幕解析度爲 280 × 160 點，x，y 座標分別爲 0 到 279 和 0 到 159（方向同低解析度）

指令

```
HPLOT x1,y1
HPLOT x1,y1 TO x2,y2
HPLOT x1,y1 TO x2,y2 TO … TO xn,yn
```

分別畫一點，一向量及 n 段向量。

高解析度的顏色指令爲

```
HCOLOR=n
```

n 爲 0 到 7 之整數。各數代表的顏色列於下：

0	黑	4	黑
1	綠	5	不定
2	藍	6	不定
3	白	7	白

其中 5 和 6 視顯示器種類而定，另外上表出現兩種黑白的詳細解釋請參閱參考書籍(3)。

同樣的高解析度繪圖螢幕通常預留四行文字區。此時可用指令

```
POKE -16302,0
```

將文字區除去而將 y 值擴充到 0～191。

上述相反效果可用下列指令達到。

```
POKE -16301,0
```

指令

```
HGR2
```

將保留高解析度第一頁記憶區不動而見螢幕轉換到顯示第二頁記憶區，並容許使用者用前面說過的各種畫圖指令在此頁上畫圖。要再轉換回第一頁，使用指令

```
POKE -16300,0
```

再轉換回第二頁，則使用指令

```
POKE -16299,0
```

HGR 及HGR 2 指令在轉換前先將其代表的記憶頁清除，所以如果你想交換顯示出兩頁的圖形，就不能使用 HGR 及 HGR 2 指令。

```
POKE 240,32    and    POKE 240,64
```

兩指令不論正在顯示何種螢幕，都可用來使 HPLOT 指令對第一頁或第二頁做畫圖工作，同時不影響另一頁的內容。

指令

```
TEXT
```

重新回到文字模式。開機時或由指令 TEXT 執行後而到文字模式時，所有的資料都存在文字第一頁記憶區，而當 HGR 轉換到高解析繪圖時，並不會破壞此記憶區，並可用下列指令重回到文字區而不影響內容

```
POKE -16303,0
```

同時 Apple 也有低解析繪圖及文字模式的第二記憶頁，這些都是用來做特殊應用的，詳情請參見參考書籍(3)。

對於圖形表的建立及畫圖的有關指令，如 DRAW，XDRAW，ROT 及 SCALE 等敍述都已在第 3-5 及 3-6 節中討論過。

8—8 使用磁碟機

對文字檔做讀與寫是磁碟作業系統中最簡單的工作。通常需要重覆使用的大量資料都必須寫成磁片的文字檔裡。另外如果你自己寫出一些有價值的程式，也可保存在磁片上。

開文字檔以做讀或寫的準備工作可用指令

```
PRINT CHR$(4);"OPEN";file-name
```

其中CHR$(4)代表CONTROL-D的ASCII字元，送此字元到系統是通知它將輸出管道由監督程式轉到磁碟作業系統去，後接的字串則視爲對作業系統的指令，所以開一檔後的指令應爲：

```
PRINT CHR$(4);"command-word";file-name
```

此處" command-word "代表"WRITE"，"READ"或"DELETE"等指令，其分別代表寫入此檔，讀出此檔及刪去此檔的功能。其中讀或寫的資料項則分別放在後面的 READ 或 PRINT 敍述裡。

關閉一檔的指令爲

```
PRINT CHR$(4);"CLOSE";file-name
```

要將磁碟的輸入資料同時顯示在螢幕上，我們可在執行這些動作前用

指令

```
CHR$(4); PRINT "MON C I O"
```

其中C，I及O分別使作業系統的指令，輸入及輸出等資料顯示在螢幕上。這三項選擇可獨立使用或合併使用，且順序不同。NOMON 指令則將此監督功能停止。

上述各項有關 DOS 及其他如隨取檔，文字檔等詳細結構請參閱參考書籍(4)。

本書有關將數個程式連在一起的技巧（如PLOT 1，PLOT 2 及 PLOT 3）等也請參閱上述相關書籍。

H. K. $ 25.00

Apple II 家用電腦繪圖

郭金銘 譯

專業出版社